

KENDALI ROBOT BERDASARKAN PERINTAH INISIALISASI AWAL PENGGUNA

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Adit Ilham Nugroho
NIM: 135150300111043



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

KENDALI ROBOT BERDASARKAN PERINTAH INISIALISASI AWAL PENGGUNA

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Adit Ilham Nugroho
NIM: 135150300111043

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 16 Januari 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Mochammad Hannats Hanafi I., S.ST, M.T
NIK: 201405 881229 1 001

Dosen Pembimbing II

Rizal Maulana, S.T., M.T., M.Sc
NIK: 201607 891009 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

IDENTITAS TIM PENGUJI

1. Penguji 1 / Ketua Majelis2

Wijaya Kurniawan, S.T, M.T

NIP. 19820125 201504 1 002

2. Penguji 2

Bayu Rahayudi , S.T, M.T

NIP. 19740712 200604 1 001



PERNYATAAN ORISINALITAS

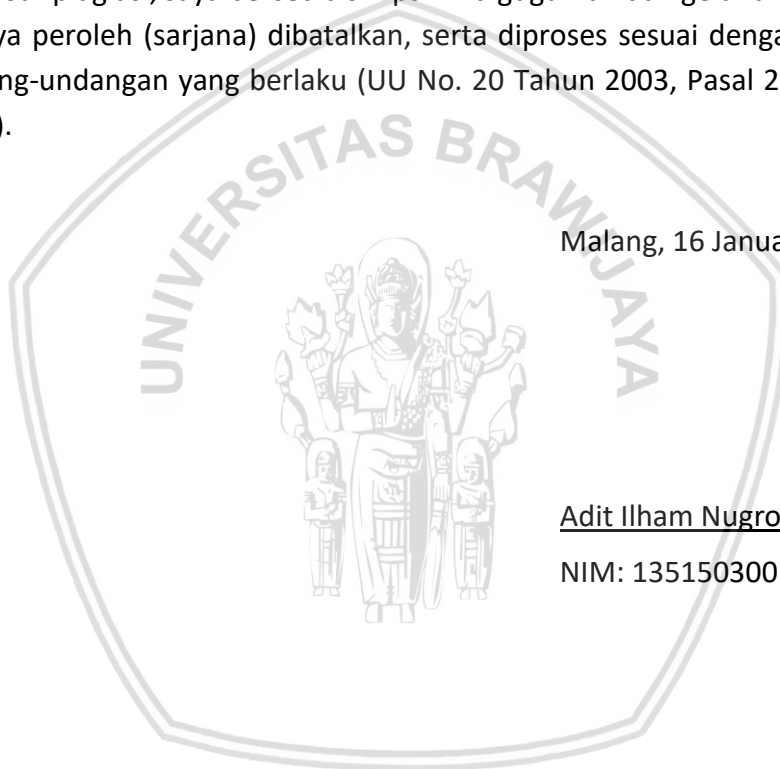
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 Januari 2018

Adit Ilham Nugroho

NIM: 135150300111043



DAFTAR RIWAYAT HIDUP

DATA PRIBADI

Nama : Adit Ilham Nugroho
 Tempat, Tgl Lahir : Kupang, 15 Februari 1995
 Jenis Kelamin : Laki-laki
 Agama : Islam
 Kewarganegaraan : Indonesia
 Status : Belum Kawin
 Alamat : Jalan Sastra Kencana 2 Blok V2 Nomor 21 Sektor XII.5,
 RT.005, RW.005, Kelurahan Ciater, Kecamatan Serpong,
 BSD, Kota Tangerang Selatan
 Telephone : -
 Hp/WA : 087809656242
 Email : aditilham152@gmail.com

PENDIDIKAN FORMAL :

2001 – 2007 SDI CIKAL HARAPAN I
 2007 – 2010 SMP Negeri 8 Tangerang Selatan
 2010 – 2013 SMA Negeri 7 Tangerang Selatan

KATA PENGANTAR

Puji syukur peneliti panjatkan atas kehadiran Tuhan Yang Maha Esa, karena limpahan rahmat dan petunjuk-Nya peneliti dapat menyelesaikan skripsi yang berjudul “Kendali Robot Berdasarkan Perintah Inisialisasi Awal Pengguna”.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan moral dan materiil yang diberikan dari berbagai pihak, maka peneliti mengucapkan banyak terima kasih kepada

1. Orang tua dan keluarga penulis yang selalu memberikan doa, motivasi, kasih sayang serta dukungan moril dan materiil sebagai penyemangat dalam menyelesaikan skripsi ini
2. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku ketua Program Studi Teknik Komputer Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya
4. Bapak Mochammad Hannats Hanafi I., S.ST, M.T dan Bapak Rizal Maulana, S.T., M.T., M.Sc. yang telah memberikan dukungan dan bimbingannya kepada peneliti untuk segera menyelesaikan skripsi ini.
5. Teman-teman penulis, Rizqy Maulana, Ayu Samura, Ivana Agustina, Oktaviany Setyowati, Nashir Umam Hasbi, Putri Laras Rinjani, Wirafadhil, Falachudin, Vatikan, Agung, Dhani, Rofiq, Boleng, Dewan, Dimas Alif Gibran, Ivan Pradito, M. Andre, Defmarshal, Denis Zulkarnain, yang banyak membantu memberikan motivasi dan semangat kepada penulis.
6. Teman-teman tercinta program studi Teknik Komputer angkatan 2013, 2012 dan 2014 yang selalu memberikan doa dan semangat dalam proses pengerjaan skripsi.

Peneliti menyadari bahwa dalam laporan ini masih banyak kekurangan. Oleh sebab itu, peneliti berharap adanya pengembangan lebih lanjut oleh pihak-pihak terkait. Semoga laporan ini dapat memberikan manfaat dan referensi untuk melakukan penelitian sebagai langkah penyempurnaan sistem.

Malang, 16 Januari 2018

Penulis

aditilham152@gmail.com

ABSTRAK

Adit Ilham Nugroho, KENDALI ROBOT BERDASARKAN PERINTAH INISIALISASI AWAL PENGGUNA

Pembimbing: Mochammad Hannats Hanafi I., S.ST, M.T dan Rizal Maulana, S.T., M.T., M.Sc.

Selama 30 tahun terakhir, penelitian robotika dan perkembangan robotika terus berkembang, robot *autonomous* yang dulu disebut robot industri tradisional dan *animal like* atau *humanoid robot*. Salah satu dari perkembangan tersebut adalah teknologi robot industri atau disebut juga *industrial robot*. Ada beberapa alasan mengapa robot yang diartikulasikan banyak dikembangkan antara lain robot ini masuk kedalam robot pabrik yang dimana industri saat ini terus berkembang dan juga memiliki ruang gerak yang bebas 360°. Robot industri dapat meningkatkan produktivitas dan efisiensi pada sebuah pabrik untuk menjalankan lini produksi. Penulis membuat sebuah *prototype* robot yang dapat bergerak sesuai dengan inisialisasi awal pengguna, robot ini memudahkan pengguna dalam menjalankan robot jika ingin merubah gerakan robot, dibandingkan dengan robot *autonomous* yang sudah ada. Robot ini tergolong robot manual yang menekan biaya pembelian robot. Pada penelitian ini hanya menggunakan *software* Arduino dengan fungsi EEPROM untuk menyimpan gerakan yang telah diberikan. *Software* Arduino merupakan perangkat lunak *open-source* untuk memudahkan penulisan kode. Alat bantu hardware Arduino tipe Mega 2560, Arduino Mega 2560 merupakan perangkat keras pengembangan mikrokontroler berbasis Arduino dengan menggunakan chip ATmega2560 untuk membuat prototipe penelitian. Penelitian ini menggunakan total 5 joint dengan menggunakan penggerak Servo, Servo Motor yang digunakan adalah Servo SG90. Pengujian gerakan servo 0°-180° memiliki nilai rata-rata error sebesar 1.85 % dan pengujian gerakan servo 0°-360° memiliki nilai rata-rata error sebesar 2.32 %. Lalu pada analisis perintah 1 gerakan nilai rata-rata error hanya sebesar 2.9 % dan pada analisis perintah keseluruhan nilai rata-rata error hanya sebesar 1.5 %.

Kata kunci: robot, servo, EEPROM, Arduino Mega.

ABSTRACT

Adit Ilham Nugroho, KENDALI ROBOT BERDASARKAN PERINTAH INISIALISASI AWAL PENGGUNA.

Lecturer: Mochammad Hannats Hanafi I., S.ST, M.T and Rizal Maulana, S.T., M.T., M.Sc.

Over the past 30 years, robotics research and robotics developments continue to evolve, autonomous robots that used to be called traditional industrial robots and animal like or humanoid robots. One of these developments is industrial robot technology or also called industrial robot. There are several reasons why many articulated robots are developed, among others, this robot into the robot factory where the current industry continues to grow and also has a free space 360 °. Industrial robots can increase productivity and efficiency in a factory to run the production line. The author created a prototype robot that can move according to the initial initialisation of the user, this robot allows users to run the robot if you want to change the robot movement, compared with existing autonomous robot. This robot is classified as a manual robot that reduces the cost of purchasing a robot. In this study only use Arduino software with EEPROM function to save the movement that has been given. Arduino software is an open-source software for easy code writing. Arduino hardware device type Mega 2560, Arduino Mega 2560 is a hardware development of Arduino-based microcontroller using ATmega2560 chip to create research prototype. This study uses a total of 5 joints using Servo drives, Servo motors used are Servo SG90. Testing servo movement 0°-180° has an average error rate of 1.85% and servo movement 0°-360° has an average error rate of 2.32%. Then in order analysis 1 movement the average value of error is only 2.9% and in the overall command analysis the average error rate is only 1.5%.

Keywords: robot, servo, EEPROM, Arduino Mega.

DAFTAR ISI

PENGESAHAN	ii
IDENTITAS TIM PENGUJI	iv
PERNYATAAN ORISINALITAS	ivv
DAFTAR RIWAYAT HIDUP	ivv
KATA PENGANTAR.....	vii
ABSTRAK.....	viii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 Mikrokontroler	6
2.2.2 Servo Motor	7
2.2.3 <i>Push Button</i>	9
2.2.4 <i>Toggle Switch</i>	11
2.2.5 EEPROM	12
2.2.6 <i>Debounce</i>	12
BAB 3 METODOLOGI	13
3.1 Alur Metode Penelitian.....	13
3.2 Studi Literatur	14
3.3 Rekayasa Kebutuhan.....	14

3.3.1	Kebutuhan Fungsional	14
3.3.2	Kebutuhan Non-Fungsional	14
3.4	Perancangan.....	14
3.5	Implementasi	15
3.6	Pengujian dan Analisis.....	15
3.7	Kesimpulan.....	15
BAB 4	REKAYASA KEBUTUHAN.....	16
4.1	Gambaran Umum Sistem.....	16
4.2	Analisis Kebutuhan Sistem.....	16
4.2.1	Kebutuhan Fungsional	16
4.2.2	Kebutuhan Non-Fungsional	17
4.3	Batasan Desain Sistem.....	18
BAB 5	PERANCANGAN DAN IMPLEMENTASI	19
5.1	Gambaran Umum Sistem.....	19
5.2	Perancangan Sistem.....	19
5.2.1	Perancangan Sistem Perangkat Keras	20
5.2.2	Perancangan Sistem Perangkat Lunak.....	23
5.3	Implementasi Sistem	31
5.3.1	Implementasi Sistem Perangkat Keras	32
5.3.2	Implementasi Sistem Perangkat Lunak.....	355
BAB 6	PENGUJIAN DAN ANALISIS.....	41
6.1	Pengujian Ketepatan Pembacaan Servo	41
6.1.1	Tujuan Pengujian	41
6.1.2	Prosedur Pengujian.....	41
6.1.3	Hasil dan Analisis Pengujian.....	41
6.2	Pengujian Perintah 1 Gerakan	47
6.2.1	Tujuan Pengujian	47
6.2.2	Prosedur Pengujian.....	47
6.2.3	Hasil dan Analisis Pengujian.....	47
6.3	Pengujian Perintah Keseluruhan	50
6.3.1	Tujuan Pengujian	51
6.3.2	Prosedur Pengujian.....	51

6.3.3 Hasil dan Analisis Pengujian.....	51
BAB 7 PENUTUP	54
7.1 Kesimpulan.....	54
7.2 Saran	54
DAFTAR PUSTAKA.....	56



DAFTAR TABEL

Tabel 2.1 Spesifikasi Mikrokontroler Arduino Mega 2560	7
Tabel 2.2 Spesifikasi Servo SG90	9
Tabel 2.3 Spesifikasi <i>Toggle Switch</i>	12
Tabel 5.1 Pin Skematik Keseluruhan Sistem	21
Tabel 5.2 <i>Library Code</i>	35
Tabel 5.3 Kode Program Inisialisasi Pin	36
Tabel 5.4 Kode Program Variabel <i>Global</i>	36
Tabel 5.5 Kode Program Merekam	37
Tabel 5.6 Kode Program Bergerak Sekali	38
Tabel 5.7 Kode Program Bergerak Berkali-kali	38
Tabel 5.8 Kode Program Inisialisasi Nilai Sekali	38
Tabel 5.9 Kode Program <i>Record</i>	39
Tabel 5.10 Kode Program <i>Play</i>	39
Tabel 5.11 Kode Program <i>Reset</i>	40
Tabel 6.1 Hasil Pengujian Servo 0-°180°	43
Tabel 6.2 Hasil Pengujian Servo 0-°360°	45
Tabel 6.3 Hasil Pengujian Perintah 1 Gerakan	49
Tabel 6.4 Hasil Pengujian Perintah Keseluruhan	52

DAFTAR GAMBAR

Gambar 2.1 Mikrokontroler Arduino Mega 2560	7
Gambar 2.2 Servo SG90	8
Gambar 2.3 Contoh gerakan Servo	9
Gambar 2.4 Ilustrasi gerakan Servo	9
Gambar 2.5 <i>Push Button</i>	10
Gambar 2.6 Simbol tombol tekan	10
Gambar 2.7 Ilustrasi prinsip kerja tombol tekan	11
Gambar 2.8 <i>Toggle Switch</i>	11
Gambar 2.12 Rangkaian RC.....	12
Gambar 3.1 Diagram Alir Penelitian.....	13
Gambar 3.2 Blok Diagram Sistem	15
Gambar 5.1 Gambaran Umum Sistem	19
Gambar 5.2 Skematik Keseluruhan Sistem	20
Gambar 5.3 Desain Lengan Robot.....	22
Gambar 5.4 Diagram Alir Sistem.....	24
Gambar 5.5 Diagram Alir Fungsi Merekam.....	26
Gambar 5.6 Diagram Alir Fungsi Bergerak Sekali.....	27
Gambar 5.7 Diagram Alir Fungsi Bergerak Berkali-kali	27
Gambar 5.8 Diagram Alir Fungsi <i>Recording</i>	29
Gambar 5.9 Diagram Alir Fungsi <i>Play</i>	30
Gambar 5.10 Diagram Alir Fungsi <i>Reset</i>	31
Gambar 5.11 Rangkaian RC/Rangkaian <i>debounce</i>	32
Gambar 5.12 Pemasangan Kabel <i>Feedback</i> di Servo SG90.....	32
Gambar 5.13 Bentuk Lengan Balok Terbuka.....	33
Gambar 5.14 Posisi Lengan Robot	34
Gambar 5.15 Penampang Sistem Dengan Menggunakan Akrilik	34
Gambar 5.16 Total Perancangan Sistem Perangkat Keras.....	35
Gambar 6.1 Tampilan Pada Serial Monitor	42
Gambar 6.2 Hasil Pembacaan Pada Multimeter	42
Gambar 6.3 Posisi Awal Servo	48

Gambar 6.4 Nilai Sudut Belajar	48
Gambar 6.5 Hasil Sudut Aktual	49
Gambar 6.6 Posisi Servo Setelah Dilakukan Perekaman	51
Gambar 6.7 Hasil Sudut Aktual	52



BAB 1 PENDAHULUAN

1.1 Latar belakang

Perkembangan dunia robotika terjadi sangat cepat pada era ini. Ketertarikan manusia yang tinggi terhadap perkembangan robot berujung pada penelitian yang semakin bermunculan.

Selama 30 tahun terakhir, penelitian robotika dan perkembangan robotika terus berkembang robot *autonomous* yang dulu robot industri tradisional dan *animal like* atau *humanoid robot*. Salah satu dari perkembangan tersebut adalah teknologi robot industri atau disebut juga *industrial robot*. Berdasarkan penjelasan dari *website* Administrasi Keselamatan Kerja dan Kesehatan Kerja Departemen Tenaga Kerja Amerika Serikat yang diakses pada tanggal 23 September 2017 mengatakan bahwa robot industri berarti alat mekanis multifungsi yang dapat diprogram dan dirancang untuk memindahkan material, komponen, peralatan, atau perangkat khusus melalui gerakan terprogram variabel untuk melakukan berbagai tugas. Sistem robot industri tidak hanya mencakup robot industri tetapi juga perangkat dan / atau sensor yang dibutuhkan robot untuk melakukan tugasnya serta melakukan sekuensing atau memantau antarmuka komunikasi.

Robot umumnya digunakan untuk melakukan tugas yang tidak aman, berbahaya, sangat berulang, dan tidak menyenangkan. Mereka memiliki banyak fungsi yang berbeda seperti penanganan material, perakitan, pengelasan busur, pengelasan ketahanan, pemuatan alat mesin dan fungsi pembongkaran, pengecatan, penyemprotan, dll (Department of Labor, 2017).

Articulated arm robot berdasarkan penjelasan yang telah dibuat oleh Carlos M. Gonzales, seorang *technology*. *Articulated Robots* atau *Robots Arms* dilengkapi sendi putar yang dapat berkisar dari struktur dua sendi sederhana hingga struktur rumit dengan 10 atau lebih sambungan. Lengan terhubung ke alas yang memiliki sambungan memutar. Sendi putar menghubungkan kaitan di lengan. Setiap sendi adalah sumbu yang berbeda dan memberikan tingkat kebebasan tambahan. Lengan robot industri memiliki empat atau enam sumbu. Robot semacam itu terutama digunakan untuk operasi perakitan, mesin *die-casting*, *fettling*, pengelasan gas dan busur, dan penerapan cat (Gonzales, 2016).

Ada beberapa alasan mengapa robot yang diartikulasikan banyak dikembangkan antara lain robot ini masuk kedalam robot pabrik yang dimana industri saat ini terus berkembang dan juga memiliki ruang gerak yang bebas 360°. Robot industri dapat meningkatkan produktivitas dan efisiensi pada sebuah pabrik untuk menjalankan lini produksi.

Inisialisasi adalah proses mencari dan menggunakan nilai yang ditentukan untuk data variabel yang digunakan oleh program komputer. Misalnya, sistem operasi atau program aplikasi diinstal dengan nilai default atau nilai yang ditentukan pengguna yang menentukan aspek tertentu dari bagaimana sistem atau program berfungsi. Biasanya, nilai-nilai ini disimpan dalam file inisialisasi.

Ketika sistem operasi atau program aplikasi pertama kali dimasukkan ke dalam memori, sebagian program melakukan inisialisasi - yaitu, tampilannya dalam file inisialisasi, menemukan nilai pasti untuk menggantikan nilai variabel, dan bertindak sesuai. Disini menggunakan inisialisasi awal pengguna, jadi pengguna memberikan gerakan terhadap robot dan gerakan tersebut menjadi inisialisasi untuk menjadi gerakan robot (Thing, 2016).

Penelitian yang sudah dilakukan oleh Adiyatma Ghazian ST, Ir. Nurussa'adah, MT., Mochammad Rif'an ST., MT. menjelaskan terkait gerakan robot secara koordinat menggunakan metode invers kinematik. Invers kinematik berfokus pada penentuan nilai-nilai variabel joint agar didapat posisi yang diinginkan pada orientasi *end-effector*. Didapat kan hasil pada posisi 0° membutuhkan waktu sebesar 2,1ms dan pada posisi 90° membutuhkan waktu sebesar 0,9ms. Penelitian berikutnya oleh Dr. Bindu A Thomas, Stafford Michahial, Shreeraksha.P, Vijayashri B Nagvi, Suresh M. menjelaskan tentang *autonomous* robot lengan dibidang industri. Kondisi robot tersebut akan menunggu dengan waktu yang telah ditentukan ketika ada hambatan pada saat menggerakkan objek. Dan akan kembali bergerak ketika hambatan telah dibersihkan, namun jika masih terjadi hambatan maka *buzzer* akan dihidupkan untuk memanggil operator atau personil industri untuk mengatasi masalah.

Penulis membuat sebuah *prototype* robot yang dapat bergerak sesuai dengan inisialisasi awal pengguna, robot ini memudahkan pengguna dalam menjalankan robot jika ingin merubah gerakan robot, dibandingkan dengan robot *autonomous* yang sudah ada. Robot ini tergolong robot manual yang menekan biaya pembelian robot.

Pada penelitian ini hanya menggunakan *software* Arduino. *Software* Arduino merupakan perangkat lunak *open-source* untuk memudahkan penulisan kode. Alat bantu hardware Arduino tipe Mega 2560, Arduino Mega 2560 merupakan perangkat keras pengembangan mikrokontroler berbasis Arduino dengan menggunakan chip ATmega2560 untuk membuat prototipe penelitian. Penelitian ini menggunakan total 5 joint dengan menggunakan penggerak Servo, Servo Motor yang digunakan adalah Servo SG90. Servo Motor merupakan sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Dengan total ada 5 joint dan diinisialisasi awal dari pengguna, sehingga pengguna bisa memberikan gerakan yang diinginkan.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana merancang gerak robot berdasarkan perintah inisialisasi awal pengguna?
2. Bagaimana implementasi kendali robot berdasarkan perintah inisialisasi awal pengguna?

3. Bagaimana analisis hasil gerak servo pada kendali robot berdasarkan perintah inisialisasi awal pengguna?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Merancang kendali robot berdasarkan perintah inisialisasi awal pengguna.
2. Implementasi kendali robot berdasarkan perintah inisialisasi awal pengguna.
3. Analisis hasil gerak servo pada kendali robot berdasarkan perintah inisialisasi awal pengguna.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis

- a. Menambah pengetahuan mengenai penggunaan servo dan Arduino untuk menerima perintah inisialisasi awal pengguna.
- b. Sarana penerapan dan pengembangan ilmu pengetahuan dan wawasan selama perkuliahan.
- c. Menambah pengetahuan mengenai sistem pembacaan data yang diterima oleh Arduino.

2. Bagi Perguruan Tinggi

Perwujudan dari Tri Dharma Perguruan Tinggi. Dengan dilakukan penelitian ini diharapkan dilakukan pengembangan ilmu pengetahuan dan teknologi yang dapat bermanfaat bagi masyarakat.

3. Bagi Pembaca

- a. Memberikan wawasan mengenai penggunaan servo dan Arduino untuk menerima perintah inisialisasi awal pengguna.
- b. Dapat dijadikan referensi untuk penelitian gerak tanah selanjutnya.

1.5 Batasan masalah

Ruang lingkup masalah dalam penelitian ini adalah sebagai berikut :

1. Sistem hanya untuk membaca inisialisasi awal pengguna dan menggerakkan sesuai dengan inisialisasi tersebut.
2. Hanya menggunakan total 5 servo.
3. 5 Servo hanya bergerak 90° hingga 180° .
4. Alat hanya bekerja pada pergerakan ruang kerja.

1.6 Sistematika pembahasan

Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini menjelaskan latar belakang penelitian, permasalahan-permasalahan penelitian, tujuan penelitian, manfaat yang dapat diambil dari penelitian, serta batasan masalah, dan sistematika penyajian laporan penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas landasan teori terkait yang membantu penelitian, serta menjelaskan tentang penelitian-penelitian serupa yang pernah dilakukan. Pada bagian ini dipaparkan teori-teori serta pustaka yang dipakai pada waktu penelitian.

BAB 3 METODOLOGI

Bab ini memaparkan langkah-langkah yang digunakan untuk membahas permasalahan dalam penelitian. Pada bagian ini dijelaskan alat yang digunakan untuk melakukan perencanaan dan mendapatkan spesifikasi kebutuhan robot.

BAB 4 REKAYASA KEBUTUHAN

Bab ini membahas secara rinci terkait deskripsi umum dari sistem, rekayasa antar-muka sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional, batasan desain sistem dan alur kerja sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab perancangan dan implementasi menjelaskan perancangan desain dari perangkat keras dan desain perangkat lunak yang nantinya akhir dari perancangan ini akan mendapatkan hasil yang sesuai dengan yang diharapkan dan akan diimplementasikan sesuai dengan hasil analisa dan perancangan.

BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis menjelaskan proses dan hasil dari pengujian metode yang diterapkan pada sistem untuk kemudian dilakukan analisis berdasarkan hasil dari pengujian tersebut.

BAB 7 PENUTUP

Bab penutup berisi kesimpulan dan saran yang dapat diperoleh berdasarkan pengujian dan analisis yang telah dilakukan agar sistem dapat dikembangkan lebih lanjut di kemudian hari.

BAB 2 LANDASAN KEPUSTAKAAN

Pada penelitian ini, pustaka sebagai referensi berasal dari berbagai jurnal pendukung yang berkaitan dengan koordinat sudut lengan robot dari Adiyatma Ghazian Pratama ST, Ir. Nurussa'adah, MT., Mochammad Rif'an, ST., MT.. Selanjutnya penelitian tentang robot lengan industri *autonomous* dari Dr. Bindu A Thomas, Stafford Michahial, Shreeraksha.P, Vijayashri B Nagvi, Suresh M. Sedangkan beberapa referensi tentang dasar teori sebagai pendukung untuk pembuatan sistem meliputi Arduino dan servo.

2.1 Kajian Pustaka

Penelitian dari Adiyatma Ghazian Pratama ST, Ir. Nurussa'adah, MT., Mochammad Rif'an, ST., MT. menjelaskan bahwa untuk menentukan sudut lengan robot berdasarkan koordinat yang telah dimasukkan kedalam PC dengan menggunakan metode Invers Kinematik. Masukkan koordinat ke PC melalui *User Interface* yang dibuat dari Qt. Invers Kinematik berfokus pada penentuan nilai-nilai *variable joint* agar didapat posisi yang diinginkan dan orientasi *end-effector*. Dengan menggunakan 3 servo, hasil pengujian penelitian ini berdasarkan persamaan invers kinematik didapatkan hubungan nilai waktu dengan sudut yang berbeda. Servo Lengan Atas Putar Kanan pada posisi 0° membutuhkan waktu sebesar 2,1 ms dan pada posisi 90° membutuhkan waktu sebesar 0,9 ms. Servo Ketiak Kanan pada posisi 0° membutuhkan waktu sebesar 0,9 ms dan pada posisi 90° membutuhkan TON sebesar 1,7 ms. Servo Siku Kanan pada posisi 0° membutuhkan waktu sebesar 2,1 ms dan pada posisi 90° membutuhkan waktu sebesar 1,3 ms. Dari ketiga rentang waktu pada ketiga servo yang berbeda persamaan linier konversi dari perubahan nilai sudut terhadap nilai TOP-nya juga berbeda.

Pada penelitian dari Dr. Bindu A Thomas, Stafford Michahial, Shreeraksha.P, Vijayashri B Nagvi, Suresh M. menjelaskan bahwa lengan robot adalah manipulator robotik yang terprogram dengan fungsi yang sama seperti lengan manusia. Biasanya manusia mengambil atau bergerak tanpa memikirkan langkah-langkah yang terlibat. Tidak dengan robot atau lengan robot untuk mengangkat atau memindahkan sesuatu, seseorang harus memberitahunya untuk melakukan beberapa tindakan dalam urutan tertentu. Dari menggerakkan lengan hingga memutar sendi lengan, sehingga kita bisa mengendalikan sendi. Makalah ini menyajikan 3 lengan robot otomatis gabungan yang bisa digunakan di industri untuk melakukan tugas kontinu seperti memindahkan barang dari konveyor ke tempat lain, sensor digunakan untuk mendeteksi hambatan yang ada pada saat menjalankan tugas. Jika ada kendala saat menggerakkan objek, lengan akan menunggu hingga waktu yang telah ditentukan untuk pembersihan objek. Jika hambatan dibersihkan, lengan akan melanjutkan pekerjaannya. Jika masih ada hambatan, maka buzzer akan dihidupkan sehingga personil dari industri bisa mengatasi masalah dan menghapus hambatannya.

Pada penelitian ini, peneliti mengangkat permasalahan kendali robot di dunia industri, yaitu pada pergerakan fleksibilitas lengan robot yang sering kali menggunakan autonomik robot. Autonomik robot ini tidak fleksibel karena hendak merubah pergerakan atau membuat gerakan baru harus mengganti kode yang berada pada lengan robot, sehingga hal ini menyebabkan kurang efisiennya waktu dalam proses pengerjaan.

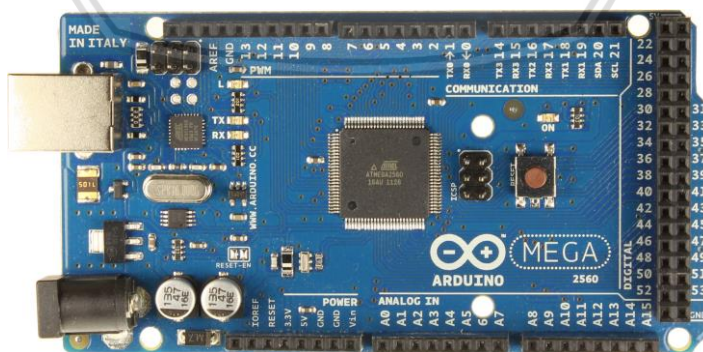
Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, maka pada penelitian ini akan menjelaskan tentang memberikan gerak yang diinginkan oleh awal pengguna dan akan bergerak seperti yang sudah diberikan oleh pengguna. Dengan adanya penelitian ini, diharapkan hasil akhir penelitian dapat memberikan solusi rekayasa pergerakan robot industri.

2.2 Dasar Teori

Pada dasar teori, berisi penjelasan dan menguraikan beberapa teori dan literature tentang komponen, serta aplikasi pendukung pembuatan sistem ini.

2.2.1 Mikrokontroler

Mikrokontroler adalah komputer mikro dalam satu chip tunggal. Mikrokontroler memadukan CPU, ROM, RWM, I/O paralel, I/O seri, counter-timer, dan rangkaian clock dalam satu chip.¹ Salah satu contoh mikrokontroller yaitu Arduino. Arduino Mega adalah platform hardware open source yang di rancang dengan menggunakan 8-bit Atmel AVR-mikrokontroler. Atau 32-bit ARM dengan kecepatan clock 16 MHz. Arduino memiliki sebuah interface USB, 54 pin I/O digital, 16 input analog, sebuah input power, ICSP header dan tombol reset. Pada board Arduino Mega, mikrokontroler dapat dinyalakan dengan USB laptop dengan kabel serial atau dengan menggunakan AC to DC Adapter atau baterai tegangan 6-20 V (Park, 2014). Bentuk fisik mikrokontroller Arduino Mega 2560 ditunjukkan pada Gambar 2.1.



Gambar 2.1 Mikrokontroler Arduino Mega 2560

Sumber : arduino.cc (2017)

Berikut ini merupakan spesifikasi dari Mikrokontroler Arduino Mega 2560

Tabel 2.1 Spesifikasi Mikrokontroler Arduino Mega 2560

Sumber : ecadio.com (2017)

Chip mikrokontroller	ATmega2560
Tegangan operasi	5V
Tegangan input (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM output
Analog Input pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori Flash	256 KB, 8 KB telah digunakan untuk bootloader
SRAM	8 KB
EEPROM	4 KB
<i>Clock speed</i>	16 Mhz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g

2.2.2 Servo Motor

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanent motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan (KOLLMORGEN, 2016).

Secara umum terdapat 2 jenis motor servo. Yaitu motor servo standard dan motor servo *Continuous*. Servo motor tipe standar hanya mampu berputar 180 derajat. Motor servo standard sering dipakai pada sistim robotika misalnya untuk membuat "Robot Arm" (Robot Lengan). sedangkan Servo motor *continuous* dapat berputar sebesar 360 derajat. motor servo *Continuous* sering dipakai untuk Mobile Robot. Pada badan servo tertulis tipe servo yang bersangkutan.

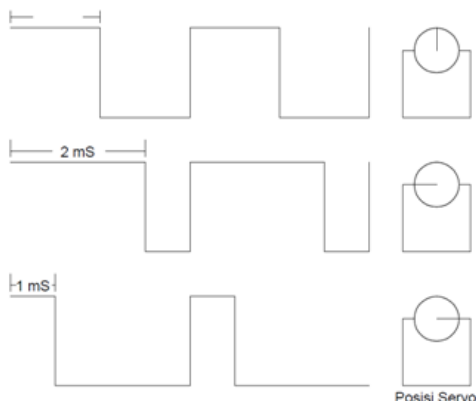
Motor servo merupakan sebuah motor dc kecil yang diberi sistim gear dan potensiometer sehingga dia dapat menempatkan "horn" servo pada posisi yang dikehendaki. Karena motor ini menggunakan sistim close loop sehingga posisi "horn" yang dikehendaki bisa dipertahankan. Bentuk fisik Servo SG90 ditunjukkan pada Gambar 2.2.



Gambar 2.2 Servo SG90

Sumber : servodatabase.com (2017)

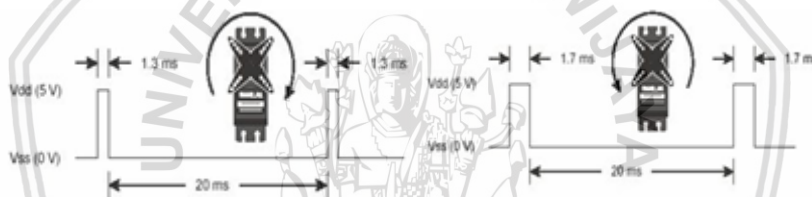
Pengendalian gerakan batang motor servo dapat dilakukan dengan menggunakan metode PWM. (Pulse Width Modulation). Teknik ini menggunakan system lebar pulsa untuk mengemudikan putaran motor. Sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada Gambar 2.3 dengan pulsa 1.5 mS pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.



Gambar 2.3 Contoh gerakan Servo

Sumber : servodatabase.com (2017)

Untuk menggerakkan motor servo ke kanan atau ke kiri, tergantung dari nilai delay yang kita berikan. Untuk membuat servo pada posisi center, berikan pulsa 1.5ms. Untuk memutar servo ke kanan, berikan pulsa $\leq 1.3\text{ms}$, dan pulsa $\geq 1.7\text{ms}$ untuk berputar ke kiri dengan delay 20ms, seperti yang telah diilustrasikan pada Gambar 2.4:



Gambar 2.4 Ilustrasi gerakan Servo

Sumber : servodatabase.com (2017)

Berikut ini merupakan Spesifikasi Servo SG90

Tabel 2.2 Spesifikasi Servo SG90

Sumber : indo-ware.com (2017)

Kecepatan Beban	0,12 detik / 60° (4.8V)
<i>Stall Torque</i>	1,6 kg / cm (4.8V)
Suhu Operasi	-30 ~ +60° Celcius
Set Mati	7 mikrodetik
Tegangan Operasi	4.8V - 6V
Bekerja Saat Ini	kurang dari 500mA
Panjang Kabel	180mm

2.2.3 Push Button

Tombol tekan adalah bentuk saklar yang paling umum dari pengendali manual yang dijumpai di industri. Tombol tekan NO (*Normally Open*) menyambung

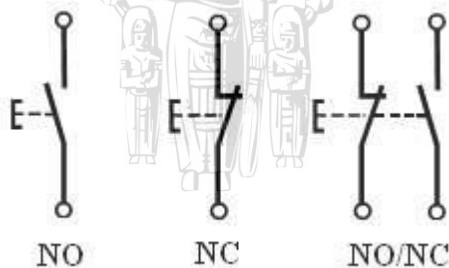
rangkaian ketika tombol ditekan dan kembali pada posisi terputus ketika tombol dilepas. Tombol tekan NC (*Normally Closed*) akan memutus rangkaian apabila tombol ditekan dan kembali pada posisi terhubung ketika tombol dilepaskan. Bentuk fisik Push Button dapat dilihat pada Gambar 2.5:



Gambar 2.5 Push Button

Sumber : micros.com (2017)

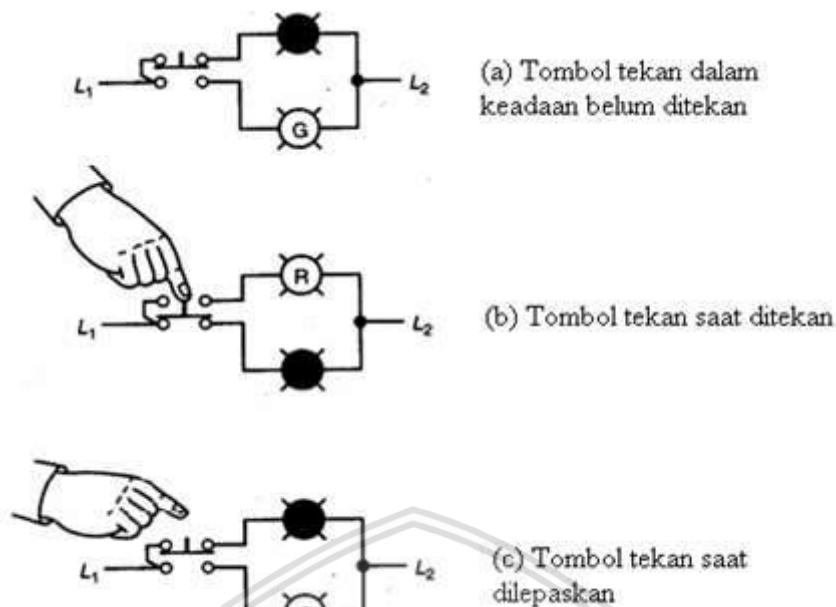
Ada juga tombol tekan yang memiliki fungsi ganda, yakni sudah dilengkapi oleh dua jenis kontak, baik NO maupun NC. Jadi tombol tekan tersebut dapat difungsikan sebagai NO, NC atau keduanya. Ketika tombol ditekan, terdapat kontak yang terputus (NC) dan ada juga kontak yang terhubung (NO). Beberapa bentuk tombol tekan dapat dilihat pada Gambar 2.6.



Gambar 2.6 Simbol tombol tekan

Sumber : Indonesia Australia Partnership For Skills Development (2017)

Ilustrasi prinsip kerja tombol tekan dapat dilihat pada Gambar 2.7 dibawah berikut. Pada gambar (a) tersebut diperlihatkan posisi tombol dalam keadaan belum di sentuh. Gambar (b) menunjukkan tombol tekan sedang ditekan dan gambar (c) saat tekanan pada tombol telah dilepaskan. Perbedaan fungsi masing-masing kontak dilihat dari hidup dan matinya lampu (lampu R dan G) secara bergantian. Dalam prakteknya tombol tekan difungsikan sebagai tombol untuk menjalankan rangkaian kontrol (START) dan mematikan rangkaian kontrol (STOP).



Gambar 2.7 Ilustrasi prinsip kerja tombol tekan

Sumber : Peralatan Kontrol pada Sistem Tenaga Listrik dan Pnuematik (2017)

2.2.4 Toggle Switch

Toggle switch juga merupakan saklar umum. Penjelasan sama seperti diatas bahwa saklar toggle digunakan untuk mengubah sirkuit antara terbuka atau tertutup. Berikut merupakan bentuk fisik dari *Toggle Switch*, dapat dilihat pada Gambar 2.8 :



Gambar 2.8 Toggle Switch

Sumber : micros.com (2017)

Berikut ini merupakan Spesifikasi Toggle Switch

Tabel 2.3 Spesifikasi *Toggle Switch*

Sumber : electron.com(2017)

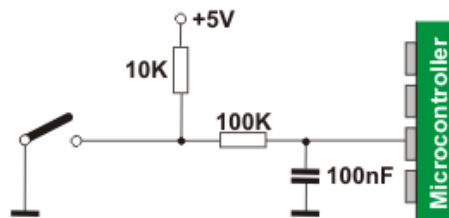
Tipe	Toggle
Fungsi <i>Switch</i>	ON-OFF
Tegangan	AC 125/ 250V
Arus	6/3A
Lubang Pemasangan	6mm
Suhu Operasional	-25-85 ° C
Ukuran	12.5x6.5x9x.5mm

2.2.5 EEPROM

EEPROM (*electrically erasable programmable read-only memory*) adalah memori read-only yang dapat dimodifikasi pengguna (ROM) yang dapat dihapus dan diprogram ulang (ditulis berulang kali) melalui penerapan voltase listrik yang lebih tinggi dari normal. Tidak seperti chip EPROM, EEPROMs tidak perlu dikeluarkan dari komputer untuk dimodifikasi. Namun, chip EEPROM harus dihapus dan diprogram ulang secara keseluruhan, tidak selektif. Ini juga memiliki kehidupan yang terbatas - yaitu, berapa kali program tersebut dapat diprogram ulang dibatasi hingga puluhan atau ratusan ribu kali. Dalam EEPROM yang sering diprogram ulang saat komputer sedang digunakan, kehidupan EEPROM bisa menjadi pertimbangan desain yang penting (Thing, 2017).

2.2.6 Debounce

Debounce merupakan rangkaian RC untuk menghilangkan *bouncing* pada *push button* dan *toggle switch*, karena jika tidak menggunakan rangkaian RC maka hasil yang didapatkan pada *push button* dan *toggle switch* akan berubah-ubah atau nilai komponen bisa bervariasi. Pada Gambar 2.12 berikut merupakan rangkaian RC untuk mencegah *bouncing* terjadi.



Gambar 2.12 Rangkaian RC

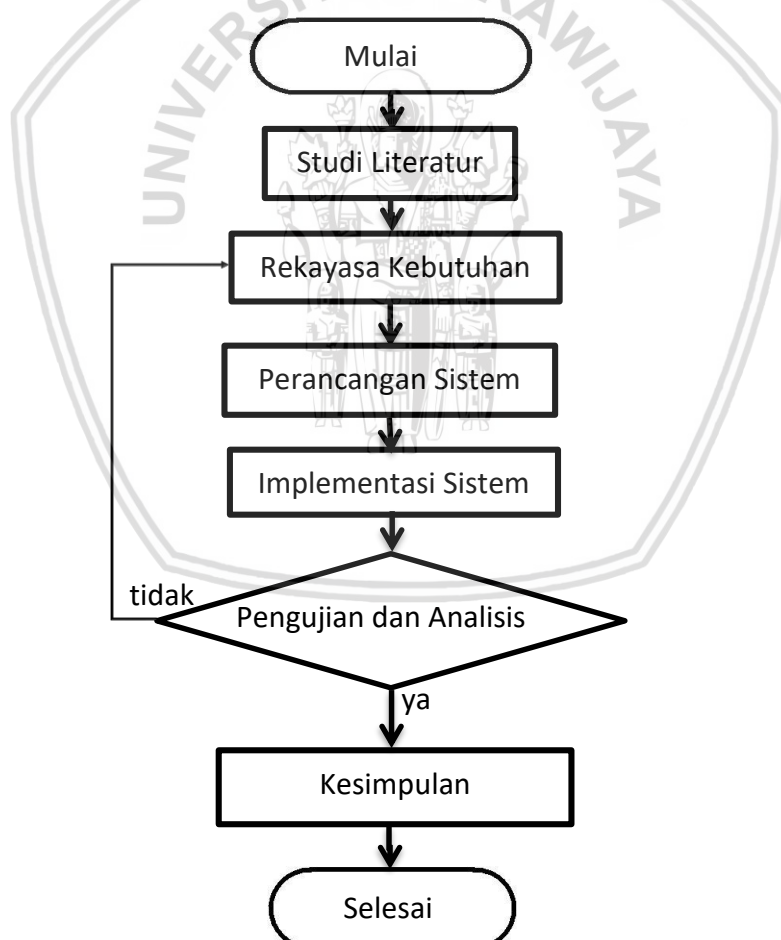
Sumber : agfi.staff.ugm.ac.id(2017)

BAB 3 METODOLOGI

Pada bab ini akan menjelaskan terkait metode dan langkah-langkah yang dilakukan dalam perancangan “KENDALI ROBOT BERDASARKAN PERINTAH INISIALISASI AWAL PENGGUNA”. Penelitian ini bersifat implementatif pengembangan.

3.1 Alur Metode Penelitian

Penelitian ini diawali dengan studi literatur yang terkait dengan tinjauan pustaka dan dasar teori. Penelitian ini bersifat implementatif karena berupa implementasi pembuatan kendali robot berdasarkan perintah inisialisasi awal pengguna menggunakan Arduino. Diawali dengan menentukan alur metode penelitian untuk digunakan sebagai langkah yang akan ditempuh untuk menyelesaikan penelitian. Alur metode penelitian yang dilakukan untuk pembuatan sistem ini dapat dilihat dari diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2 Studi Literatur

Studi literatur merupakan tahap mencari dan melakukan penyusunan teori dasar dan referensi yang mendukung sistem yang dibuat sebagai penunjang penelitian. Dalam penelitian ini dijelaskan teori-teori mengenai pergerakan *articulated arm robot*, mikrokontroler Arduino Uno, dan servo motor.

3.3 Rekayasa Kebutuhan

Rekayasa kebutuhan ditujukan untuk memenuhi segala kebutuhan yang diperlukan oleh sistem yang akan dirancang pada penelitian ini. Analisis kebutuhan akan dibagi menjadi 2 jenis kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan proses apa saja yang dikerjakan oleh sistem serta fungsi apa saja yang harus dihasilkan oleh sistem. Kebutuhan fungsional dari sistem ini diantaranya adalah:

1. Sistem dapat membaca pergerakan yang diberikan oleh pengguna.
2. Sistem dapat menyimpan pergerakan tersebut.
3. Sistem dapat bergerak sesuai dengan perintah inisialisasi awal pengguna.

3.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional menjelaskan kebutuhan apa saja yang dibutuhkan oleh sistem terkait dengan kebutuhan perangkat keras maupun perangkat lunak. Kebutuhan non-fungsional dari sistem ini diantaranya adalah:

Kebutuhan perangkat keras, yaitu:

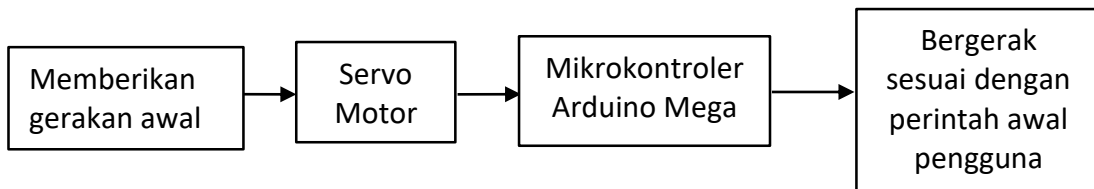
1. Arduino Mega 2560.
2. Servo Motor SG90.
3. *Push Button*.
4. *Toggle Switch* On Off

Kebutuhan perangkat lunak, yaitu:

1. Software Arduino

3.4 Perancangan

Perancangan sistem adalah tahap yang dilakukan setelah analisis kebutuhan sistem. Perancangan dilakukan apabila seluruh kebutuhan sistem telah terpenuhi. Pada tahap ini, perancangan akan dibagi menjadi beberapa tahap, yaitu perancangan perangkat keras, dimana akan membahas perancangan prototipe dari sistem yang akan dibuat dan perancangan perangkat lunak, dimana algoritma yang digunakan dalam pergerakan robot akan diterapkan sehingga sistem dapat bergerak dengan maksimal sesuai dengan perintah inisialisasi awal pengguna.



Gambar 3.2 Blok Diagram Sistem

Berdasarkan Gambar 3.2, perancangan sistem yang dibuat pada penelitian ini dijelaskan sebagai berikut.

1. Gerakan awal yang diberikan oleh pengguna merupakan gerakan manual yang akan langsung diterima oleh Servo Motor, gerakan berupa variable bebas servo direkam dan langsung menuju Mikrokontroler Arduino Mega.
2. Data yang direkam kemudian diolah mikrokontroler Arduino Mega. Untuk menjalankan gerakan tersebut pengguna menekan tombol play. Selanjutnya, robot akan bergerak sesuai dengan perintah awal pengguna.

3.5 Implementasi

Implementasi sistem dimulai dari perancangan bentuk lengan robot agar dapat bekerja sesuai yang diinginkan. Penulis juga membuat program yang nantinya akan didownload pada mikrokontroler, melakukan perancangan penerimaan dan pengiriman data di Arduino. Setelah seluruh perancangan terpenuhi, dilanjutkan proses implementasi sistem berdasarkan perancangan yang dibuat.

3.6 Pengujian dan Analisis

Pada tahap pengujian sistem, dilakukan pengujian dengan beberapa skenario yaitu pengujian pembacaan servo motor 0° - 180° dan pengujian pembacaan servo motor 0° - 360° . Pengujian pembacaan servo motor 0° - 180° dilakukan dengan cara user memberikan gerakan sudut istimewa kepada servo dan mengukur nilai analog persudut istimewa tersebut dan dibandingkan dengan nilai analog dari multi meter.

Untuk pengujian pembacaan servo motor 0° - 360° hampir sama dengan sebelumnya, akan dibandingkan nilai analog servo dengan nilai analog multimeter. Analisa dilakukan untuk mengetahui faktor-faktor yang mempengaruhi kinerja servo motor tersebut.

3.7 Kesimpulan

Kesimpulan dilakukan setelah tahapan perancangan, implementasi dan pengujian metode yang diterapkan sudah selesai. Kesimpulan diambil dari hasil pengujian dan analisis hasil. Kesimpulan ini juga merupakan jawaban dari rumusan masalah yang didefinisikan sebelumnya. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan yang terjadi serta memberikan pertimbangan untuk pengembangan selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan dijelaskan secara rinci terkait gambaran umum sistem, analisis kebutuhan fungsional dan non-fungsional sistem, kebutuhan perangkat keras dan perangkat lunak serta batasan sistem.

4.1 Gambaran Umum Sistem

Kendali robot berdasarkan perintah inisialisasi awal pengguna suatu kendali berdasarkan keinginan operator yang dapat bergerak secara kontinu atau sekali jalan. Robot akan digerakan secara manual oleh operator dalam keadaan merekam, rekaman gerakan robot berupa perubahan derajat atau perubahan perpindahan dari titik awal ke titik terakhir. Setiap lengan terdapat tombol switch yang berfungsi untuk menandakan bahwa tangan tersebut sedang berjalan fungsi merekam. Setelah mendapatkan setiap perubahan tersebut akan otomatis tersimpan di Arduino dan operator akan menjalankan switch on off untuk menjalankan hasil simpanan pergerakan dari servo. Switch on off 'Play' ada 2 tipe, yang pertama kontinu dan yang kedua adalah sekali pergerakan.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan agar seluruh kebutuhan yang diperlukan untuk membangun sistem kendali robot berdasarkan perintah inisialisasi awal pengguna ini dapat dipenuhi. Dalam melakukan analisis kebutuhan sistem ini, kebutuhan akan dibagi menjadi 2, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Berikut ini merupakan beberapa kebutuhan fungsional dari sistem yang harus terpenuhi, yaitu :

1. Merancang robot berdasarkan perintah inisialisasi awal pengguna.

Fungsi ini harus dapat merancang sebuah kendali robot berupa rancangan umum, rancangan perangkat keras dan rancangan perangkat lunak berdasarkan pergerakan yang diberikan oleh pengguna sebab itu akan menjadi awal dalam melakukan penelitian. Rancangan gerakan robot yang diberikan pengguna nanti nya akan menjadi masukkan untuk sistem.

2. Implementasi kendali robot berdasarkan perintah inisialisasi awal pengguna.

Fungsi ini berguna untuk implementasi apa yang sudah dirancang sebelumnya, yaitu dengan implementasi desain robot, implementasi komponen robot, dan implementasi perangkat lunak dari robot. Pergerakan yang sudah diberikan oleh pengguna akan disimpan karena setiap perubahan gerakan robot yang diberikan oleh pengguna nantinya akan di jalankan dengan cara menjalankan hasil simpanan gerakan.

3. Analisis hasil akhir gerakan servo.

Fungsi analisis ini nantinya akan membuktikan bahwa gerakan servo di robot benar dan sesuai dengan analisis yang telah dijalankan. Nilai yang dihasilkan oleh servo berupa nilai analog *input*. Analisis ini dilakukan setelah robot diberi pergerakan sesuai dengan inisialisasi awal pengguna lalu disimpan dan dapat berjalan sesuai dengan yang sudah diberikan oleh pengguna.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional pada sistem akan dibagi menjadi 2 kebutuhan, yaitu kebutuhan perangkat keras dan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Dibutuhkan beberapa perangkat keras dalam membuat prototipe dari sistem ini, diantaranya adalah:

1. Arduino Mega 2560

Arduino Mega 2560 merupakan mikrokontroler yang berperan sebagai inti pemrosesan dari prototipe sistem yang dibuat. Arduino Mega 2560 memiliki jumlah pin yang lebih banyak jika dibandingkan dengan Arduino Uno, Arduino Mega juga memiliki jumlah maksimal memori yang lebih banyak.

2. Servo Motor SG90

Servo Motor SG90 menjadi sendi atau penggerak diantara lengan satu dengan lengan lainnya, dalam penelitian ini menggunakan 5 Servo Motor terdiri dari 2 servo 0° - 180° dan 3 servo 0° - 360° . Penggunaan Servo 0° - 180° pada lengan pertama dan lengan kedua adalah agar tidak terhambat oleh kabel, dan penggunaan servo 0° - 360° pada lengan berikutnya adalah agar robot dapat bergerak bebas. Penggunaan Servo SG90 adalah karena torsi yang dibutuhkan sesuai dengan beban yang akan digerakan oleh servo.

3. Push Button Switch

Push Button Switch disini berperan sebagai penanda Servo mana yang akan menjalankan proses perekaman dan menjalankan proses reset. Jumlah penggunaan *push button switch* sebanyak 6 buah, 5 diantaranya berfungsi sebagai penanda pada Servo dan 1 buah untuk menjalankan fungsi reset.

4. Toggle Switch On Off

Toggle Switch On Off disini menggunakan 4 buah, berfungsi sebagai On Off alat, On Off merekam, On Off menjalankan sekali gerak, dan On Off menjalankan gerakan kontinu. *Toggle switch* dibagi menjadi 4 fungsi karena robot tidak dapat menjalankan program pada waktu yang bersamaan.

5. Baterai 12V

Baterai 12V digunakan sebagai daya tambah untuk *vcc* dan *ground* dari servo.

6. Power Bank

Power Bank disini digunakan sebagai daya dari Arduino Mega setelah kodingan di-*upload* kedalam Arduino Mega.

7. Potongan Karton Duplex

Potongan karton duplex akan menjadi lengan dari robot. Lengan kedua hingga lengan kelima memiliki ukuran yang sama dan lengan pertama akan lebih panjang daripada lengan yang lainnya. Lengan 1 lebih panjang agar putaran robot sedikit lebih luas.

4.2.2.2 Kebutuhan Perangkat Lunak

Berikut ini adalah kebutuhan perangkat lunak yang digunakan dalam prototipe dari sistem ini, yaitu:

1. Fungsi EEPROM

Fungsi EEPROM disini digunakan untuk menyimpan data nilai yang telah diberikan oleh awal pengguna, yang nantinya akan menjadi nilai masukan dan nilai yang akan dijalankan oleh robot. EEPROM hanya memiliki lebar data sebesar 8 bit maka data yang ditampung hanya dapat bernilai dari 0 hingga 255.

2. Fungsi *Record*

Fungsi *record* menjadi fungsi penting dalam pembacaan nilai ADC pada servo yang digerakan dan akan langsung disimpan kedalam EEPROM dalam bentuk nilai EEPROM dan dilakukan *mapping*.

3. Fungsi *Play*

Fungsi *play* disini berfungsi sebagai fungsi yang akan menjalankan hasil simpanan pada EEPROM. Fungsi ini membaca setiap *slot* yang ada pada EEPROM dan langsung menjalankan program yang akan menggerakan servo.

4.3 Batasan Desain Sistem

Dalam pembuatan prototipe dari kendali robot berdasarkan perintah inisialisasi awal pengguna ini terdapat beberapa batasan yang diterapkan sehingga dalam proses pembahasan, perancangan maupun implementasi dari prototipe sistem ini tidak mencakup jangkauan yang luas. Batasan-batasan dari desain prototipe sistem tersebut antara lain:

1. Prototipe menggunakan 2 servo 0° hingga 180° dan 3 servo 0° hingga 360°.
2. Prototipe hanya merekam dan bergerak sesuai rekaman inisialisasi awal pengguna.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan membahas langkah-langkah perancangan dari prototipe sistem yang akan dibuat berdasarkan rekayasa kebutuhan yang telah dijelaskan sebelumnya. Proses perancangan tersebut dapat dijelaskan hal-hal mulai dari perangkat keras dan perangkat lunaknya, hingga proses implementasi dari prototipe.

5.1 Gambaran Umum Sistem

Tahap ini menjelaskan bagaimana gambaran umum sistem yang akan dibangun, perancangan sistem baik perangkat keras hingga perangkat lunak sehingga nantinya akan membentuk suatu implementasi sistem yang telah dirancang. Gambaran umum sistem akan dijelaskan pada Gambar 5.1.



Gambar 5.1 Gambaran Umum Sistem

Dapat dilihat pada Gambar 5.1 terkait gambaran umum sistem pada penelitian ini, pertama *Toggle Switch* awal berfungsi sebagai saklar on dan off dari sistem, kedua terdapat *Toggle Switch* yang berfungsi sebagai saklar merekam, ketiga setelah merekam aktif maka terdapat *Push Button*, *Push Button* disini berfungsi sebagai pemicu servo mana yang akan merekam, keempat setelah *push button* memicu maka setiap gerakan servo akan di rekam dan terbaca oleh Arduino Mega. Setelah nilai didapat maka akan tersimpan di fungsi EEPROM Arduino Mega dan *Toggle Switch* terakhir berfungsi sebagai saklar berjalan atau bergerak, maka tadi akan bergerak sesuai dengan gerakan yang diberikan.

5.2 Perancangan Sistem

Desain pada sistem kendali ini dirancang meliputi panjang perlengan, luas penampang, desain rancang *debounce* serta penempatan letak servo. Untuk membuat robot kendali ini harus mempertimbangkan keseimbangan beban servo dan daya putar servo agar dapat bergerak secara optimal. Oleh karena itu penempatan posisi *joint* atau lengan harus ditempatkan sebaik-baiknya oleh mekanik yang mengerjakan kendali robot tersebut. Pergerakan robot akan berjalan satu persatu dimulai dari servo 1 hingga servo 5.

Selain penempatan servo tersebut mekanik juga harus mempertimbangkan bahan yang digunakan untuk *joint* atau lengan tersebut, semakin ringan bahan yang digunakan semakin ringan pula beban yang dihasilkan oleh lengan pada

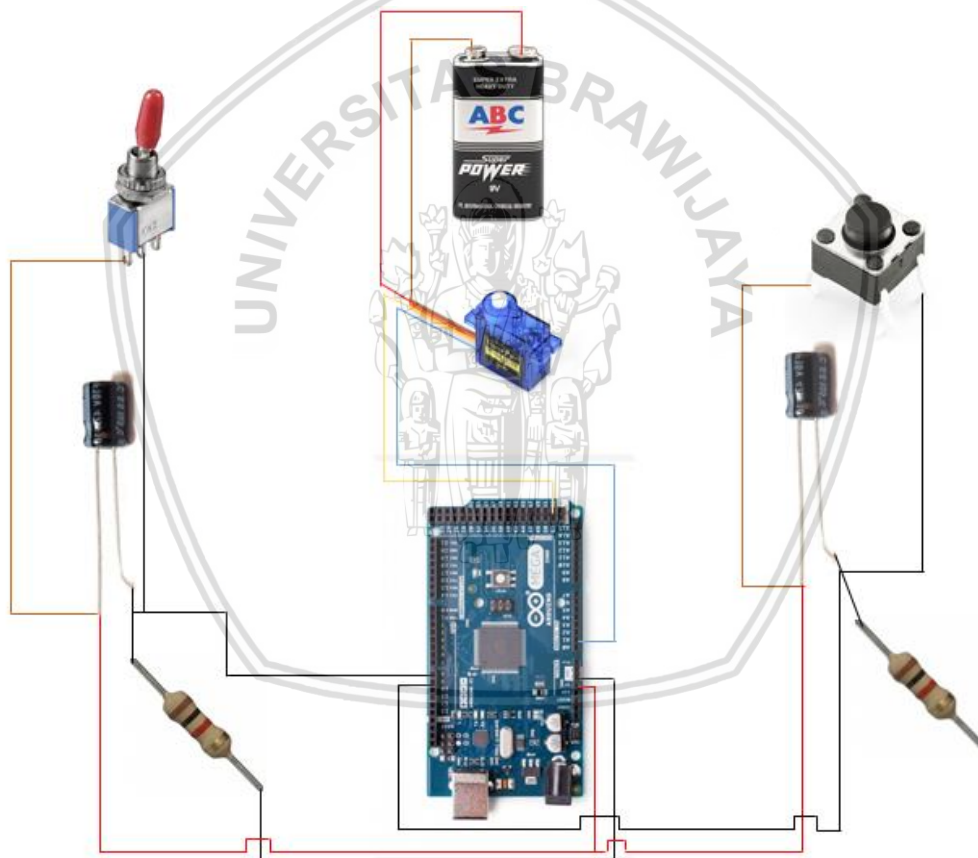
saat melakukan pergerakan yang dijalankan sesuai perintah inisialisasi awal pengguna. Perancangan robot meliputi rangkaian *debounce*, luas penampang, panjang lengan, Arduino Mega 2560, Servo SG90, *Push Button*, *Toggle Switch*.

Pada sub bab ini menjelaskan tentang perancangan sistem perangkat keras maupun perancangan perangkat lunak.

5.2.1 Perancangan Sistem Perangkat Keras

5.2.1.1 Perancangan Rangkaian Perangkat Keras

Perancangan sistem perangkat keras yang sudah dituliskan tersebut dalam bab analisis kebutuhan perangkat keras tertulis bahwa perancangan terdapat 7 komponen utama yang harus ada dalam robot, yaitu terdapat servo motor, arduino mega, *push button*, *toggle switch*, rangkaian *debounce*, baterai dan *power bank* sebagai sumber tegangan. Untuk pemasangan kabel dan komponen pada sistem ini dapat dilihat pada gambar 5.1 berikut ini:



Gambar 5.2 Skematik Keseluruhan Sistem

Pada Gambar 5.2 terdapat beberapa bagian komponen yang dijadikan 1 gambar saja, seperti servo SG90, *push button*, *toggle switch*, dan rangkaian RC. Secara keseluruhan komponen tersebut sama saja, hanya yang berbeda di pin terhadap Arduino Mega. Bisa dilihat servo menjadi ada 4 kabel karena servo telah dimodifikasi untuk bisa memberikan *feedback*. Semua servo nantinya akan terhubung kabel power ke baterai setiap pin menuju pin digital di Arduino Mega dan pin *feedback* menuju pin analog Arduino Mega.

Push button terhubung ke kapasitor 47 μ F 16V dan langsung terhubung vcc di Arduino Mega sedangkan sisi satunya terhubung ke resistor 1k ohm dan menuju pin digital, disisi sebelah resistor langsung terhubung ke *ground* Arduino Mega.

Toggle switch sistem pemasangan kabel sama seperti *push button* harus melalui rangkaian RC atau rangkaian *debounce*. Pada Tabel 5.1 berikut ini merupakan Tabel Pin Skematik Keseluruhan Sistem.

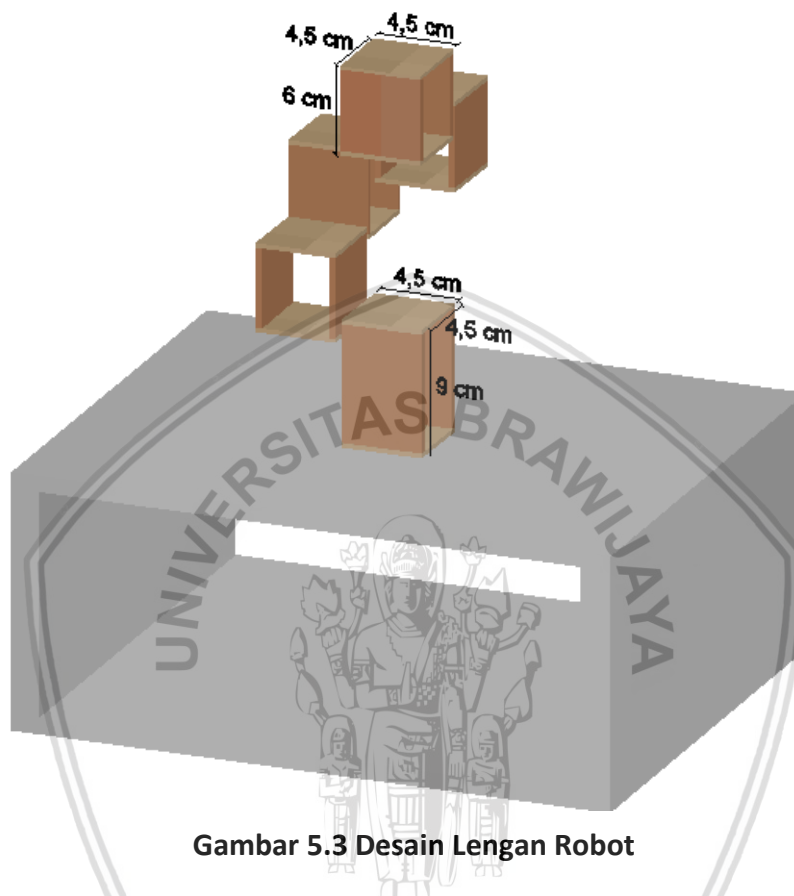
Tabel 5.1 Pin Skematik Keseluruhan Sistem

No	Arduino Mega	Servo	Push Button	Toggle Switch
1	Pin A0	Feedback Servo 1		
2	Pin A1	Feedback Servo 2		
3	Pin A2	Feedback Servo 3		
4	Pin A3	Feedback Servo 4		
5	Pin A4	Feedback Servo 5		
6	Pin 4		Push Button 1	
7	Pin 5		Push Button 2	
8	Pin 6		Push Button 3	
9	Pin 7		Push Button 4	
10	Pin 8		Push Button 5	
11	Pin 9			Merekam
12	Pin 10			Bergerak sekali
13	Pin 11			Bergerak kontinu
14	Pin 53	Pin Servo 1		
15	Pin 51	Pin Servo 2		
16	Pin 49	Pin Servo 3		
17	Pin 47	Pin Servo 4		
18	Pin 45	Pin Servo 5		
19	Pin 5V		Pin VCC	Pin VCC
20	Pin GND		Pin GND	Pin GND

5.2.1.2 Perancangan Robot Lengan

Perancangan robot lengan ini berdasarkan perhitungan panjang servo yang akan diletakan disetiap lengan. Lengan robot ini dibuat dari karton duplex yang masing-masing ukurannya adalah 6 x 4.5 cm sebanyak 8 buah, 4.5 x 4.5 cm sebanyak 10 buah, dan 9 x 4.5 cm sebanyak 2 buah. Yang nantinya lengan

pertama akan lebih panjang daripada lengan lainnya, tujuannya adalah untuk menumpu servo dengan lengan lainnya dan dapat membentuk pola setengah bola. Di setiap sisi lengan akan dipasang dengan *push button*, *push button* ini nantinya bertujuan untuk menandakan servo mana yang akan memulai sistem perekaman pergerakan. Gambar 5.3 berikut merupakan desain dari lengan robot



Gambar 5.3 Desain Lengan Robot

5.2.1.3 Perancangan Pergerakan Lengan Robot

Percangan pergerakan lengan robot dan posisi lengan ini diposisikan agar lengan tetap stabil. Nantinya lengan kedua akan diletakan disebelah kiri lengan pertama, lengan ketiga disebelah kiri lengan kedua yang berarti berada dibelakang lengan pertama, lengan keempat berada disebelah kiri lengan ketiga yang berarti membelakangi lengan kedua, lengan kelima berada disebelah kiri lengan keempat yang berarti berada diatas lengan pertama. Posisi lengan nantinya akan membentuk memutar. Pergerakan lengan robot sendiri adalah 2 servo bergerak dari 0° hingga 180° dan 3 servo bergerak dari 0° hingga 360° . Penggunaan 2 servo servo tersebut terletak pada penampang menuju lengan 1 dan lengan 2 menuju lengan 3, penempatan ini agar tidak ada kabel yang menghambat dari gerak servo tersebut. Dan penggunaan 3 servo 0° hingga 360° terletak pada sisa lengan, agar bagian atas robot dapa bergerak bebas. Sehingga membentuk gerakan setelah bola.

Langkah-langkah dalam membaca inialisasi yang pertama adalah dengan menyalakan atau memposisikan *toggle switch* merekam dalam kondisi menyala atau *on*, lalu disetiap lengan robot terdapat *push button* yang berfungsi untuk

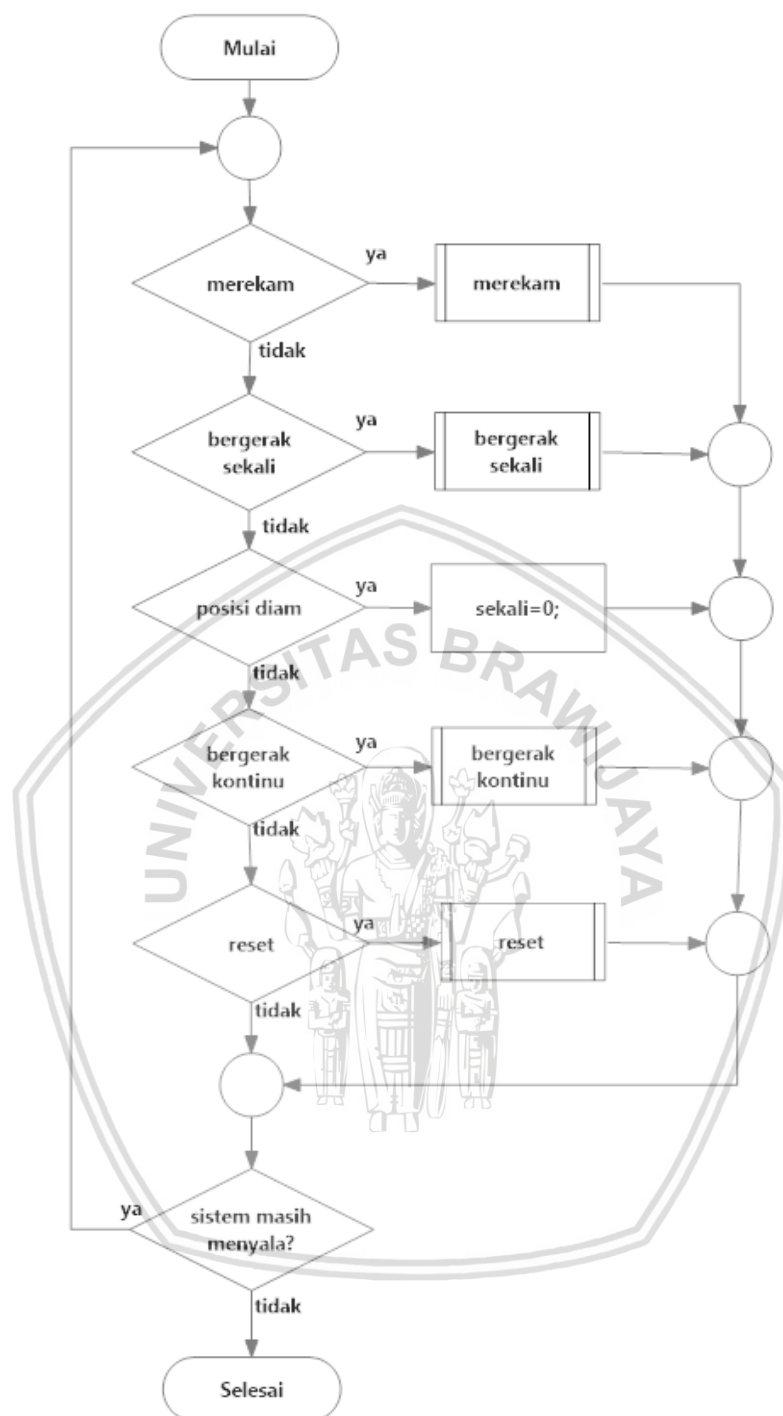
memberikan tanda bahwa servo yang lebih dulu ditekan *push button*-nya akan memulai merekam, berikutnya pengguna akan menggerakkan lengan tersebut dengan kurun waktu 5 detik dan terakhir akan tersimpan pada slot EEPROM yang telah dibagi menjadi 5 slot alamat, dengan masing-masing 50 alamat, lalu *toggle switch* dimatikan untuk memberhentikan perintah perekaman.

Untuk langkah-langkah menjalankan perintah adalah yang pertama *toggle switch* bergerak dalam kondisi menyala atau *on*, lalu program akan melakukan pembacaan pada setiap EEPROM yang telah disimpan sebelumnya, robot akan bergerak satu persatu dengan diawali dari lengan satu dan lanjut ke lengan dua dan seterusnya. Perbedaan bergerak sekali dan bergerak berkali-kali adalah ketika bergerak sekali maka alat akan bergerak dari lengan pertama hingga lengan kelima lalu berhenti, sedangkan bergerak berkali-kali maka robot akan terus bergerak dari lengan pertama hingga lengan kelima dan langsung melakukan gerakan pada lengan pertama hingga lengan kelima, bergerak berkali-kali akan berhenti ketika *toggle switch* bergerak berkali-kali dalam keadaan mati atau *off*.

Selanjutnya adalah langkah-langkah penyimpanan pada EEPROM, pertama setelah melakukan pembacaan nilai ADC (*Analog to Digital Converter*) nilai tersebut akan dilakukan *mapping* untuk mengalokasikannya pada EEPROM, satu lengan akan diberi alamat sebesar 50, sehingga membutuhkan 250 alamat pada EEPROM. Pembacaan tersebut akan dilakukan 50 alamat / 5 detik.

5.2.2 Perancangan Sistem Perangkat Lunak

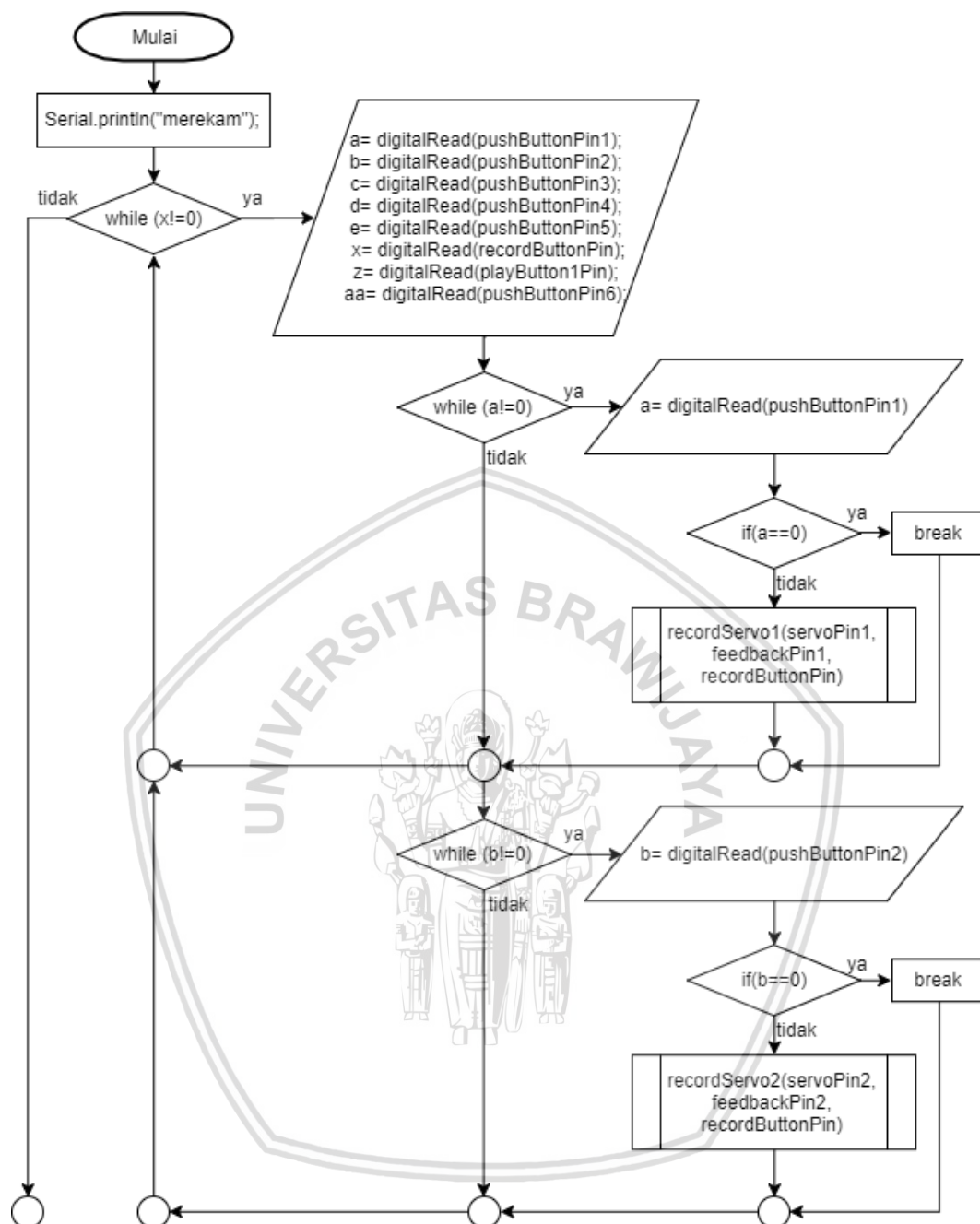
Perancangan sistem perangkat lunak yang telah ditulis di bab kebutuhan sistem perangkat lunak adalah tentang bagaimana cara agar sistem ini mampu bergerak seimbang dengan sistem perangkat keras yang akan menjalankan instruksi yang dibutuhkan sistem untuk bergerak sebagaimana fungsi yang telah ditentukan. Perangkat lunak yang digunakan dalam perancangan sistem yaitu Arduino. Perancangan perangkat lunak dijelaskan untuk mengetahui alur kerja sistem melalui diagram alir yang mana diagram tersebut berisi tentang cara perekaman inisialisasi kemudian bergerak sekali sesuai inisialisasi dan bergerak secara kontinu sesuai dengan inisialisasi. Bisa dilihat pada Gambar 5.4 merupakan diagram alir sistem.

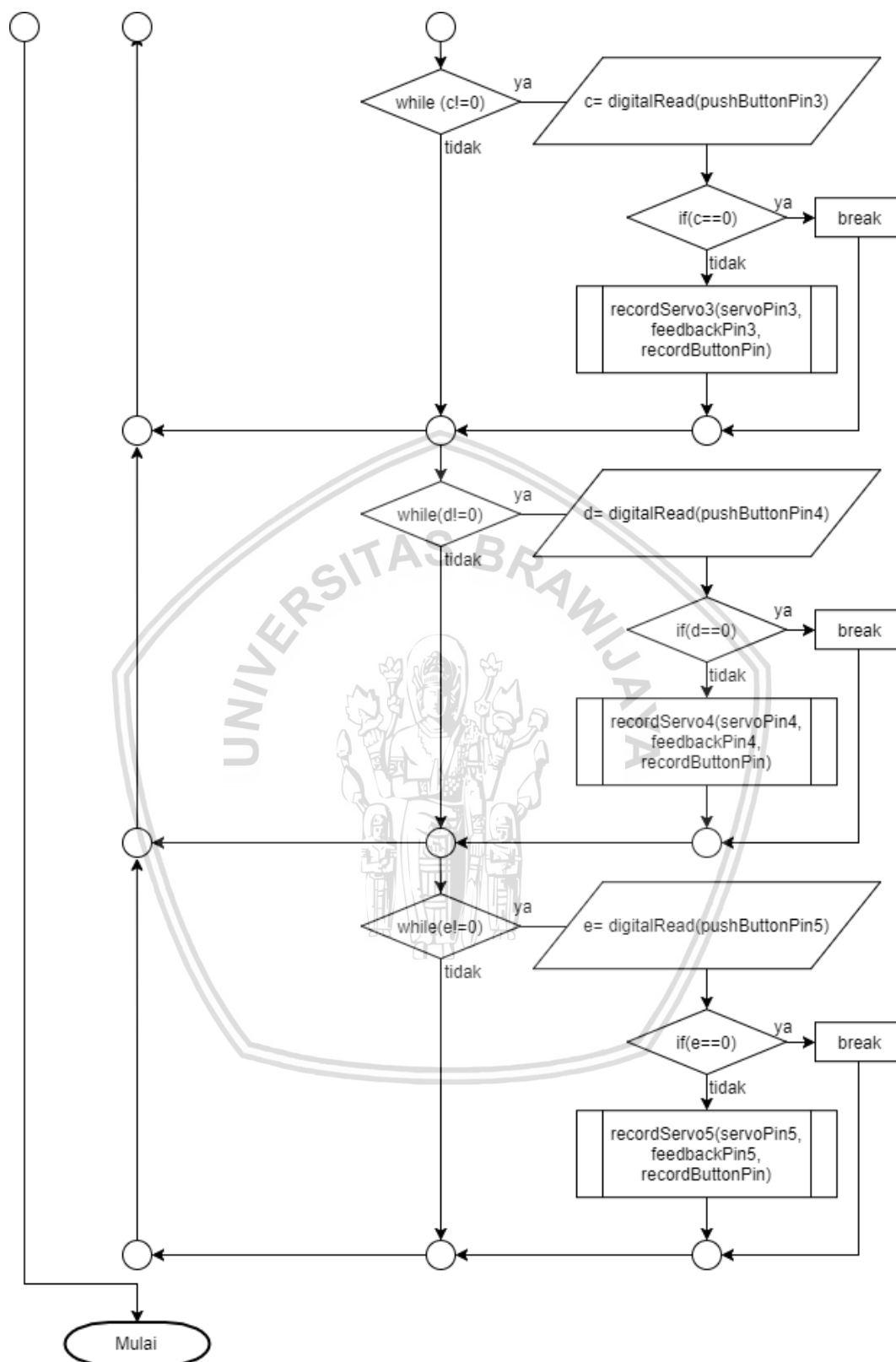


Gambar 5.4 Diagram Alir Sistem

5.2.2.1 Perancangan Fungsi Merekam

Sistem memiliki beberapa fungsi, diantaranya yaitu fungsi, diantaranya yaitu fungsi merekam, bergerak sekali, dan bergerak kontinu. Fungsi merekam terdiri dari beberapa proses yang dijelaskan pada Gambar 5.5.





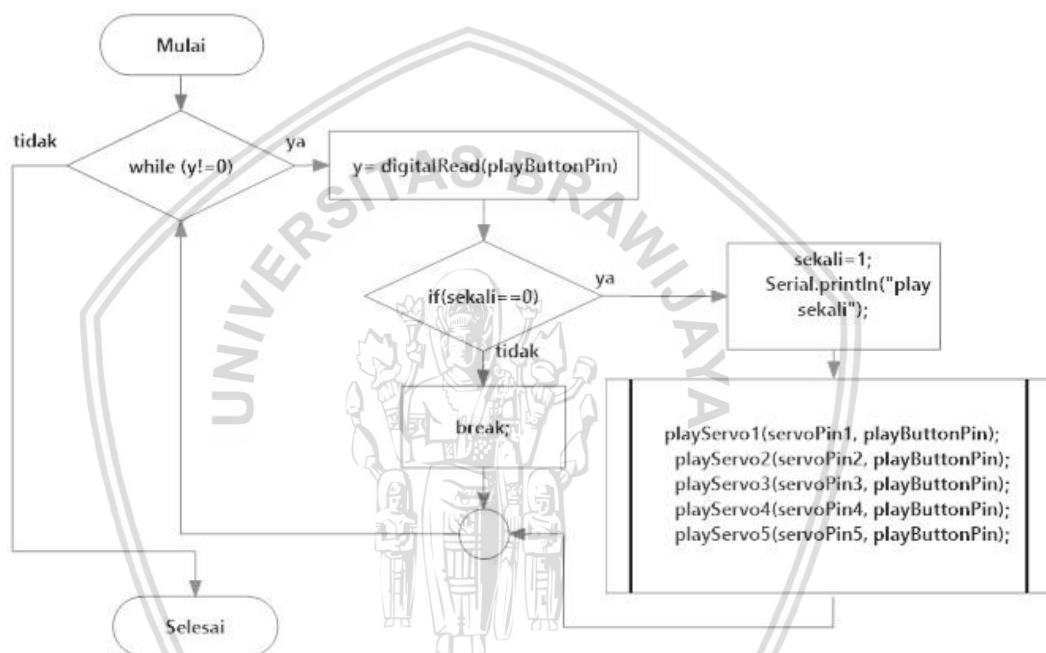
Gambar 5.5 Diagram Alir Fungsi Merekam

Tahapan pertama adalah membaca nilai dari inialisasi pada *push button*, setelah itu memasuki tahap permisalan kondisi *if-else*. Ketika a atau *push button*

pertama ditekan maka dia akan langsung membaca nilai dari a. Setelah itu memasuki permissalan kondisi jika, jika a bernilai 0 maka otomatis akan berhenti, tetapi ketika a bernilai 1 maka akan memasuki fungsi selanjutnya yaitu pemanggilan method *recoding*. Setelah selesai menjalankan method *recording* selanjutnya akan kembali ke kondisi ketika dan kembali membaca kondisi *push button* pada kondisi terbaru. Begitu pula kepada *push button* 2, 3, 4, dan 5.

5.2.2.2 Perancangan Fungsi Bergerak Sekali

Fungsi selanjutnya merupakan fungsi untuk menjalankan perintah yang sudah diinisialisasikan dan sudah disimpan kedalam EEPROM. Fungsi ini akan berjalan ketika nilai x atau inialisasi fungsi bergerak sekali bernilai 1. Fungsi ini berkerja sesuai dengan diagram alir pada Gambar 5.6 berikut.

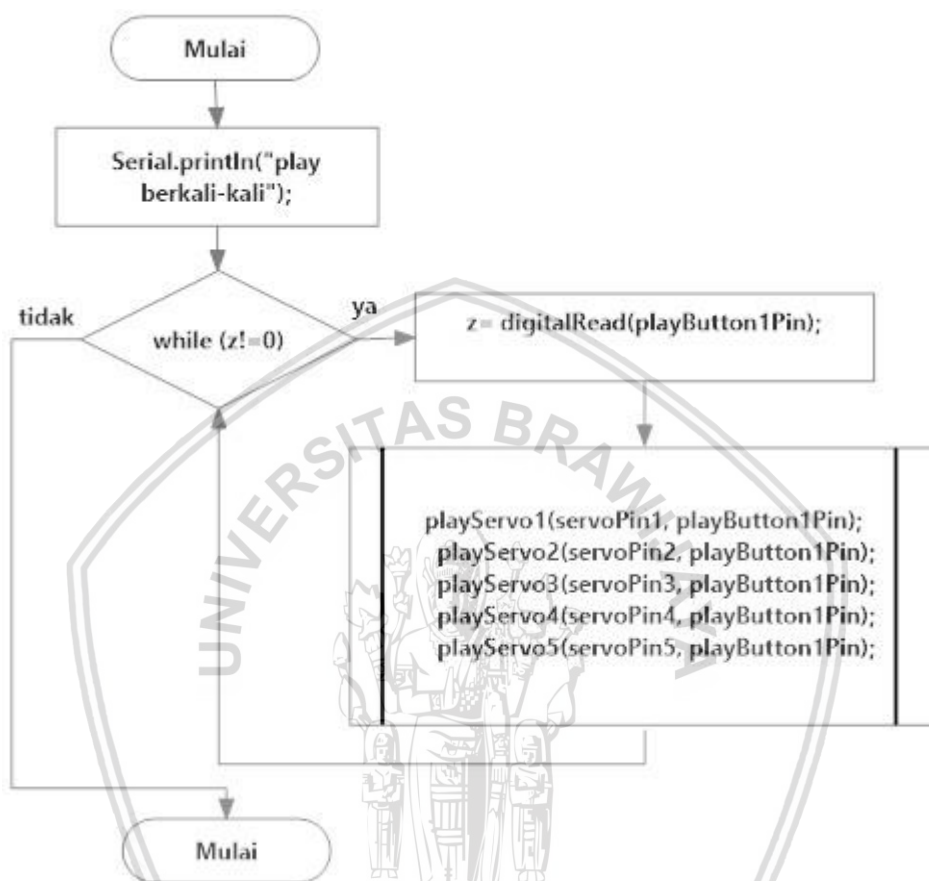


Gambar 5.6 Diagram Alir Fungsi Bergerak Sekali

Fungsi ini menginisialisasikan bahwa bergerak sekali adalah y. ketika y bernilai tidak sama dengan 0 maka akan menuju fungsi pembacaan nilai baru dari y, setelah itu menuju fungsi berikutnya yaitu permissalan kondisi jika, jika sekali bernilai 0 maka akan menuju fungsi proses yaitu menyimpan nilai baru dari sekali, maka sekali akan bernilai 1 dan menuju fungsi pemanggilan, yaitu memanggil method *play*. Sudah dijelaskan penulis sebelumnya jika nilai dari sekali akan kembali bernilai 0 jika kondisi semua *toggle switch* bernilai 0. Dapat dilihat pada kode programbergerak sekali bahwa servo akan bergerak dimulai pada servo pertama lanjut hingga servo kelima. Hal ini dilalukan karena kode program tidak dapat berjalan secara bersamaan.

5.2.2.3 Perancangan Fungsi Bergerak Berkali-kali

Fungsi berikutnya adalah fungsi bergerak berkali-kali atau gerak kontinu. Sama seperti bergerak sekali, fungsi ini akan menjalankan perintah inisialisasi dari pengguna ketika selesai merekam. Fungsi ini bekerja sesuai dengan diagram alir pada Gambar 5.7 berikut.

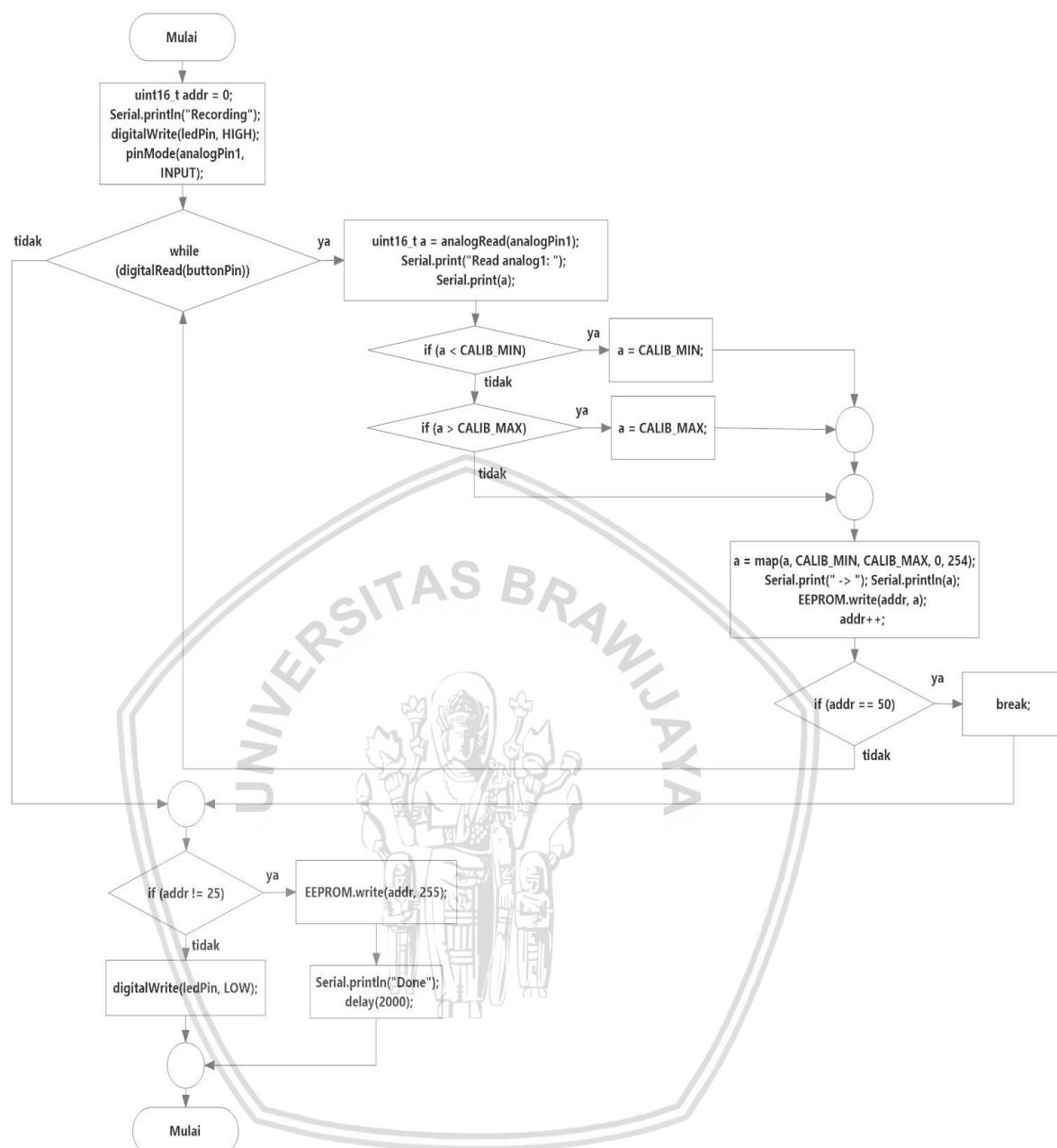


Gambar 5.7 Diagram Alir Fungsi Bergerak Berkali-kali

Perbedaan dengan fungsi bergerak sekali adalah dibagian pembentukan inisialisasi baru. Di fungsi berikut hanya akan terus bergerak jika nilai dari *toggle switch* bernilai 1. Dan otomatis berhenti ketika *toggle switch* bernilai 0. Sama halnya dengan bergerak berkali-kali, kode program bergerak berkali-kali juga hanya dapat menggerakkan servo satu persatu dimulai dari servo pertama hingga seterusnya.

5.2.2.4 Perancangan Fungsi *Recording*

Fungsi ini berguna untuk menyimpan nilai yang telah diberikan oleh awal pengguna yang akan disimpan di EEPROM. Fungsi ini berbentuk method baru yang bias dipanggil. Fungsi ini berkerja sesuai dengan diagram alir pada Gambar 5.8 berikut.



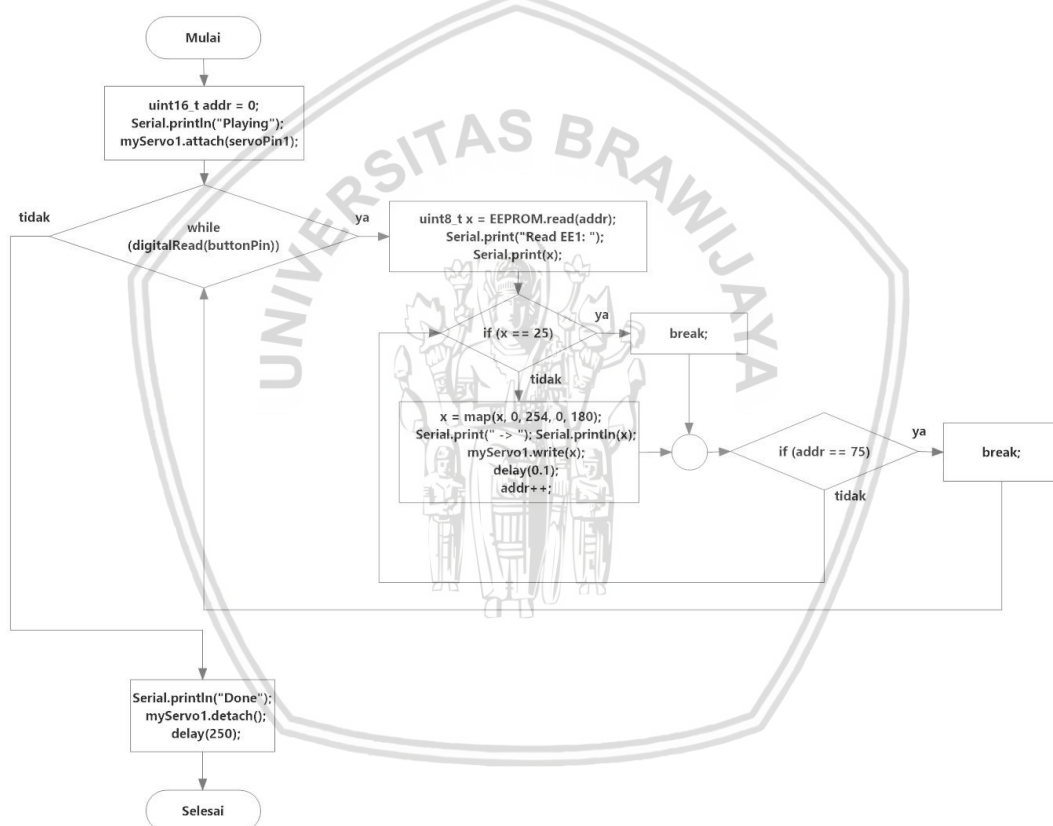
Gambar 5.8 Diagram Alir Fungsi *Recording*

Fungsi ini ketika terpanggil akan membaca *pinMode* yaitu *analogPin*. Lalu masuk ke fungsi permisalan kondisi ketika. Ketika nilai *buttonPin* bernilai sesuai maka akan menuju fungsi pembacaan kembali *analogPin* lalu akan melakukan pencetakan nilai *a*. Setelah itu akan masuk ke permisalan kondisi jika, jika *a* kurang dari kalibrasi minimal maka nilai *a* menjadi nilai kalibrasi minimal, begitu pula dengan kalibrasi maksimal. Setelah itu menuju fungsi *mapping* dan penyimpanan pada EEPROM. Fungsi ini akan terus berulang hingga nilai alamat sudah 50. Setelah itu akan menjalankan fungsi permisalan kondisi jika, jika nilai alamat tidak sama dengan 25 maka akan menyimpan nilai EEPROM berupa alamat terakhir dan nilai perubahan. Diagram Alir ini sebagai contoh pada Servo 1, penggunaan diagram alir berlaku pada servo 2 hingga servo 5. Yang membedakan hanya pada nilai awal

dan nilai akhir *address* dan iniliasisasi variabel *analogRead*, pada fungsi atau *method record* servo 1 dimulai dari *address* 0, servo 2 dimulai dari *address* 51, servo 3 dimulai dari 101, servo 4 dimulai dari 151, dan servo 5 dimulai dari 201. Untuk nilai akhir *address* pada servo 1 diakhiri nilai 50, servo 2 diakhiri nilai 100, servo 3 diakhiri nilai 150, servo 4 diakhiri nilai 200, dan servo 5 diakhiri nilai 200. Untuk inisialisasi pada *analogRead* servo 1 berupa variabel a, servo 2 berupa variabel b, servo 3 berupa variabel c, servo 4 berupa variabel d, dan servo 5 berupa variabel e.

5.2.2.5 Perancangan Fungsi *Play*

Fungsi *play* merupakan method yang dapat dipanggil untuk menjalankan perintah yang sudah disimpan pada EEPROM. Fungsi ini bekerja sesuai dengan diagram alir pada Gambar 5.9 berikut.

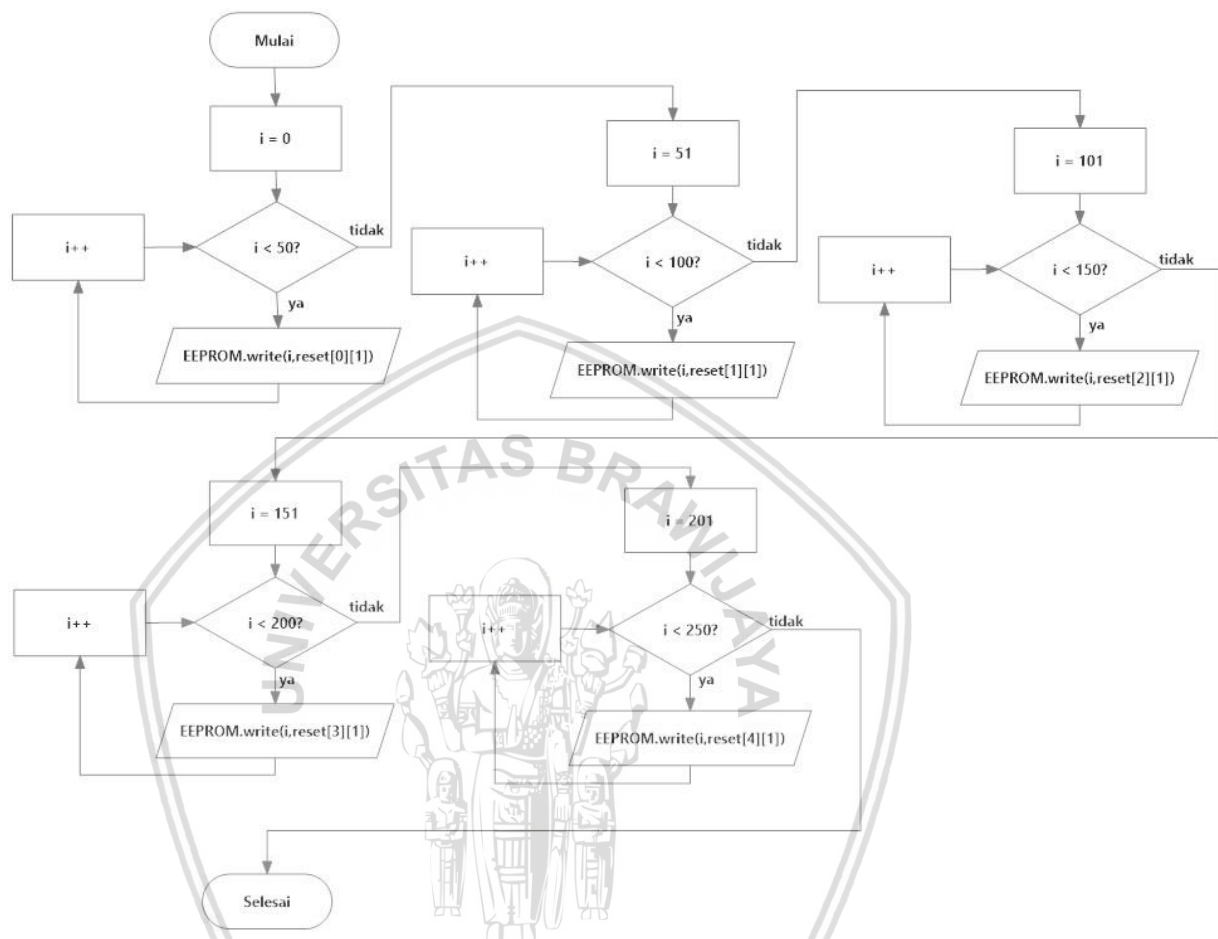


Gambar 5.9 Diagram Alir Fungsi *Play*

Fungsi ini hampir sama seperti fungsi *recording* fungsi ini berjalan hanya untuk membaca apa yang sudah ditulis atau direkam dan disimpan di EEPROM. Pembacaan tersebut peralamat dan sesuai dengan hasil *recording*. Diagram alir ini merupakan diagram alir pada servo 1, yang membedakan hanya inisialisasi variabel pada *EEPROM.read(addr)*, nilai *address* awal, dan nilai *address* akhir yang akan dibaca pada EEPROM dan dijalankan pada servo.

5.2.2.6 Perancangan Fungsi *Reset*

Fungsi *Reset* merupakan fungsi yang berguna untuk mereset gerakan, fungsi reset ini menyimpan gerakan terakhir lengan yang nantinya menjadi nilai baru pada EEPROM. Gambar 5.10 merupakan Diagram Alir Fungsi *Reset*



Gambar 5.10 Diagram Alir Fungsi *Reset*

Fungsi ini menggunakan *for* sebagai pengulangan peresetan data pada EEPROM dengan menggunakan *Array*. Fungsi *Array* akan menyimpan gerakan terakhir dari perekaman. Nilai *i* pertama ditujukan untuk servo pertama, dan akan dilakukan pengulangan pada nilai *i* hingga nilai mencapai nilai akhir *address* pada servo 1, sama seperti servo 2 hingga servo 5.

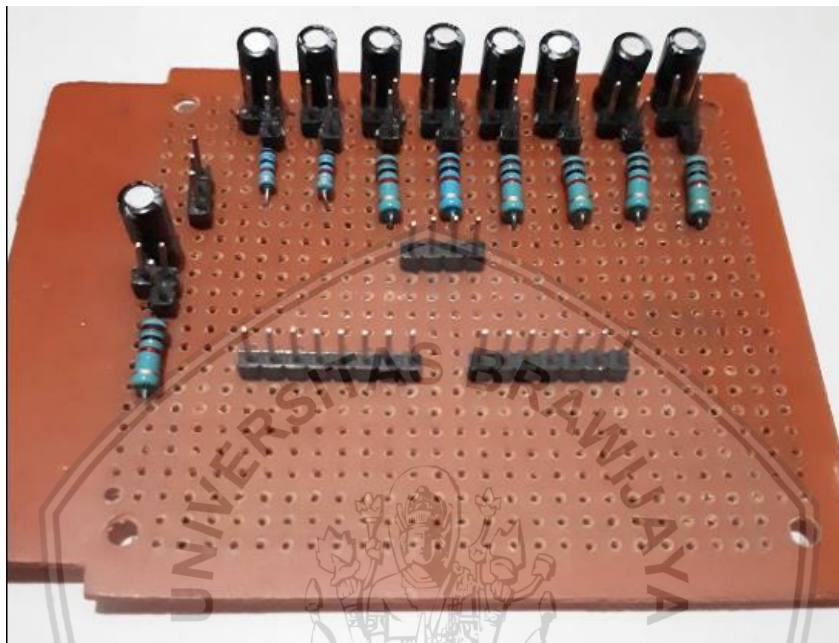
5.3 Implementasi Sistem

Pada tahap ini akan dijelaskan mengenai implementasi dari kebutuhan perancangan yang telah dibahas sebelumnya. Pada tahap ini akan dibahas terkait bagaimana sistem perangkat keras dan perangkat lunak mampu bekerja dan menghasilkan bentuk fisik maupun bentuk non fisik serta akan dijelaskan mengenai spesifikasi perangkat keras.

5.3.1 Implementasi Sistem Perangkat Keras

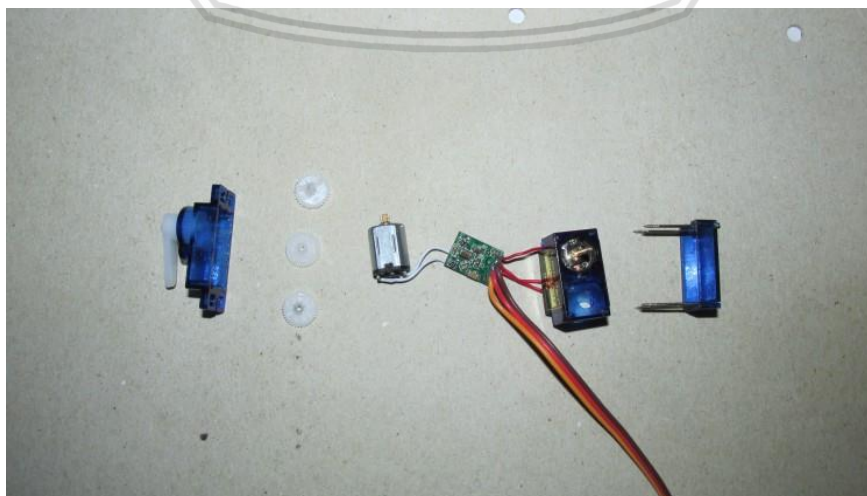
5.3.1.1 Implementasi Rangkaian Perangkat Keras

Sebelum memulai perakitan, mekanik membuat rangkaian RC atau rangkaian *debounce* yang berfungsi untuk membuat nilai dari *push button* dan *toggle switch* stabil atau tidak berubah-ubah nilai variabelnya. Dapat dilihat pada Gambar 5.11 berikut adalah pemasangan rangkaian *debounce*:



Gambar 5.11 Rangkaian RC / Rangkaian *debounce*

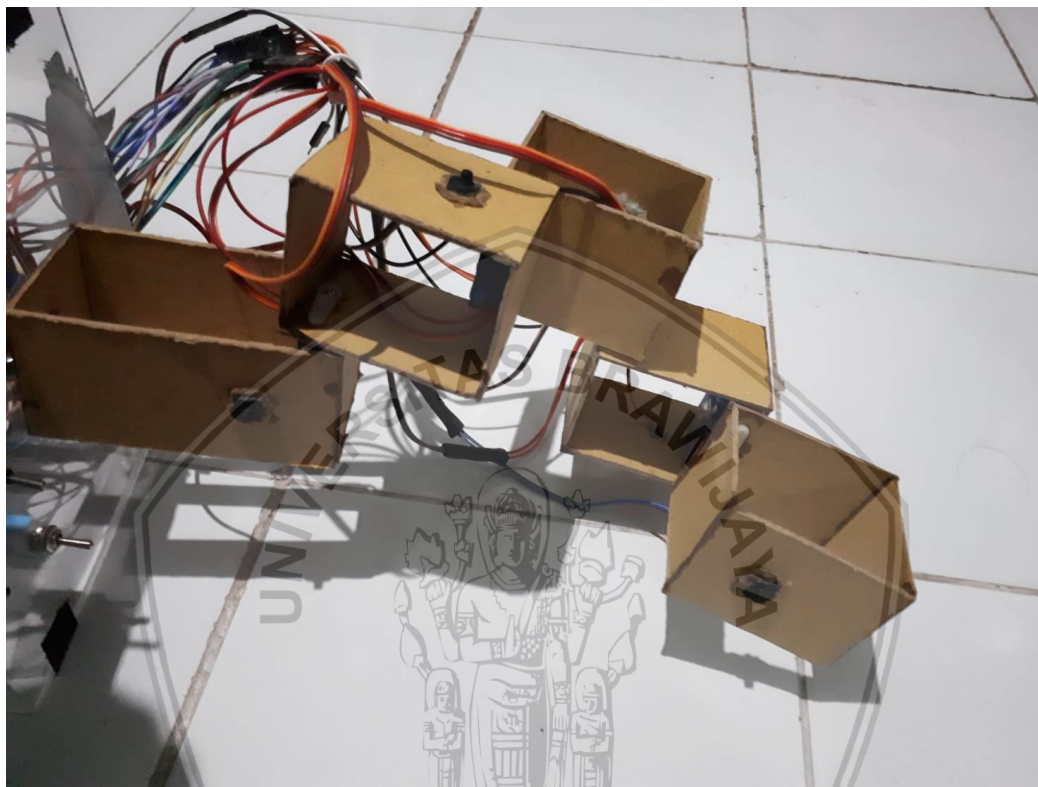
Selanjutnya melakukan modifikasi pada Servo SG90. Modifikasi ini dilakukan agar mendapatkan *feedback* dari servo itu sendiri. *Feedback* ini nantinya menjadi media untuk mengirimkan nilai untuk menggerakkan servo. Pada Gambar 5.12 berikut ini merupakan pembongkaran Servo SG90 untuk mencari *feedback*. Pencarian *feedback* sendiri menggunakan multimeter.



Gambar 5.12 Pemasangan Kabel *Feedback* di Servo SG90

5.3.1.2 Implementasi Robot Lengan

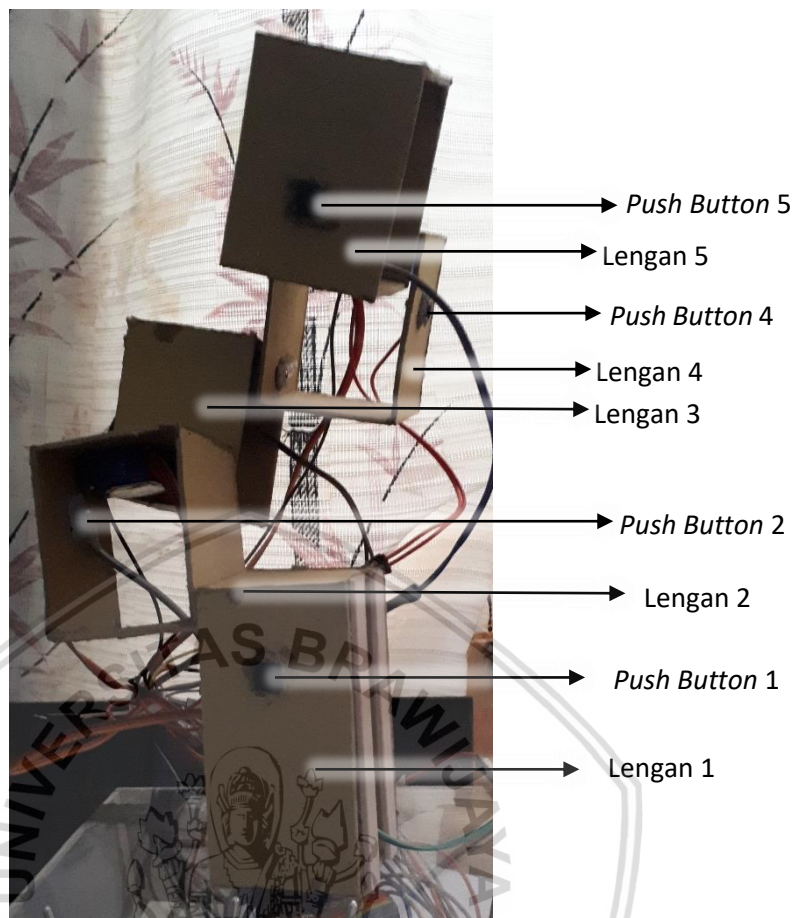
Setelah melakukan modifikasi pada servo, setelah itu melakukan implementasi lengan robot menggunakan karton duplex, lengan berbentuk balok terbuka, dengan perhitungan 6 x 4.5 cm dengan 4.5 x 4.5 cm sebanyak 4 buah dan 9 x 4.5 cm dengan 4.5 x 4.5 cm sebanyak 1 buah. Seperti Gambar 5.13 berikut ini merupakan bentuk balok terbuka dan sudah dipasang servo serta *push button*.



Gambar 5.13 Bentuk Lengan Balok Terbuka

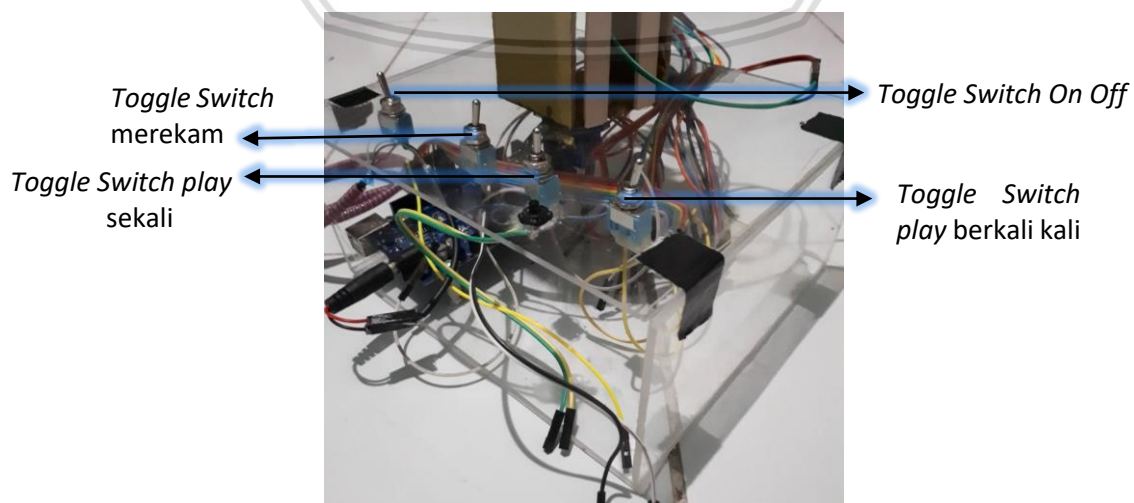
5.3.1.3 Implementasi Pergerakan Lengan Robot

Posisi lengan robot yang telah dirancang maka diimplementasikan dan akan membentuk seperti pada Gambar 5.14. Pada gambar dapat dilihat bahwa posisi lengan robot berbentuk memutar. Dan pergerakan robot akan membentuk setengah bola dengan kondisi tertentu. Pada setiap lengan robot telah dipasangi dengan *push button* dan pemasangan lengan tersebut berada pada sisi sebelah kiri setiap lengan.



Gambar 5.14 Posisi Lengan Robot

Setelah membuat posisi lengan, dilanjutkan dengan pembuatan penampang menggunakan akrilik dengan ukuran 18 x 15 cm dan sisi penampang 7 x 15 cm. Seperti pada Gambar 5.15 berikut merupakan penampang yang telah dipasang dengan 4 buah *toggle switch* dan sudah terpasang dengan lengan.



Gambar 5.15 Penampang Sistem Dengan Menggunakan Akrilik

Gambar 5.16 dibawah ini merupakan hasil dari total perancangan sistem perangkat keras yang sudah dijelaskan diatas



Gambar 5.16 Total Implementasi Sistem Perangkat Keras

5.3.2 Implementasi Sistem Perangkat Lunak

Implementasi perangkat lunak membutuhkan adanya *library* pada programnya agar memudahkan saat pembuatan dan implementasinya. *Library* pada program ini membantu untuk komponen perangkat keras dapat bekerja atau berjalan dengan baik seperti pembacaan servo, penyimpanan data di EEPROM, dan pembacaan data yang telah disimpan. Implementasi *library code* keseluruhan dapat dilihat pada Tabel :

Tabel 5.2 Library Code

1	#include <servo.h>
2	#include <EEPROM.h>
3	#define CALIB_MAX 512
4	#define CALIB_MIN 100
5	#define SAMPLE_DELAY 25

5.3.2.1 Kode Program Inisialisasi Pin

Pada bagian ini, menjelaskan pin-pin yang digunakan pada Arduino Mega. Untuk pushButtonPin1 hingga pushButtonPin5 menggunakan pin 4 hingga 8, servoPin1 hingga servoPin5 menggunakan alamat pin masing-masing. Bisa dilihat pada Tabel 5.3 berikut.

Tabel 5.3 Kode Program Inisialisasi Pin

1	uint8_t pushButtonPin1 = 4;
2	uint8_t pushButtonPin2 = 5;
3	uint8_t pushButtonPin3 = 6;
4	uint8_t pushButtonPin4 = 7;
5	uint8_t pushButtonPin5 = 8;
6	uint8_t recordButtonPin = 9; //merekam
7	uint8_t playButtonPin = 10; //play sekali
8	uint8_t playButton1Pin = 11; //play berulang
9	uint8_t pushButtonPin6 = 12; //reset
10	uint8_t servoPin1 = 53;
11	uint8_t servoPin2 = 51;
12	uint8_t servoPin3 = 49;
13	uint8_t servoPin4 = 47;
14	uint8_t servoPin5 = 45;
15	uint8_t feedbackPin1 = A0;
16	uint8_t feedbackPin2 = A1;
17	uint8_t feedbackPin3 = A2;
18	uint8_t feedbackPin4 = A3;
19	uint8_t feedbackPin5 = A4;
20	int reset[5][2];
21	int var1=1,var2=1,var3=1,var4=1,var5=1;

5.3.2.2 Kode Program *Global Variabel*

Fungsi dari kode program ini adalah variabel yang dapat digunakan atau dipanggil dalam setiap prosedur. Seperti pada Tabel 5.4 berikut ini yang menjadi *variable global* adalah sebagai berikut. A merupakan *push button* pertama, b merupakan *push button* kedua, dan seterusnya. X, y, z merupakan *toggle switch* yang fungsinya adalah fungsi merekam, bergerak sekali, dan bergerak berkali-kali. Dan terakhir adalah servo, servo 1 hingga servo 5 menjadi *variable global* karena nanti akan digunakan disetiap method.

Tabel 5.4 Kode Program Variabel *Global*

1	int a;
2	int b;
3	int c;
4	int d;
5	int e;
6	int x,y,z;
7	int sekali;
8	Servo myServo1;
9	Servo myServo2;
10	Servo myServo3;
11	Servo myServo4;
12	Servo myServo5;

5.3.2.3 Kode Program Merekam

Fungsi kode program berikut adalah untuk menjalankan fungsi merekam, fungsi merekam akan berjalan ketika fungsi tersebut bernilai 1 dilanjutkan dengan kondisi ketika *push button* mendapatkan nilai 1 lalu memanggil method *record*. Kode program berikut ini masuk kedalam method void loop. Dapat dilihat pada Tabel 5.5 berikut.

Tabel 5.5 Kode Program Merekam

1	if (x==1) {
2	Serial.println("merekam");
3	delay(10);
4	while (x!=0){
5	delay(20);
6	a= digitalRead(pushButtonPin1); //pembacaan nilai pada a
7	b= digitalRead(pushButtonPin2); //pembacaan nilai pada b
8	c= digitalRead(pushButtonPin3); //pembacaan nilai pada c
9	d= digitalRead(pushButtonPin4); //pembacaan nilai pada d
10	e= digitalRead(pushButtonPin5); //pembacaan nilai pada e
11	x= digitalRead(recordButtonPin); //pembacaan nilai pada x
12	z= digitalRead(playButtonPin); //pembacaan nilai pada z
13	aa= digitalRead(pushButtonPin6); //pembacaan nilai pada aa
14	
15	while (a!=0){
16	a= digitalRead(pushButtonPin1);
17	if(a==0) break;
18	recordServo1(servoPin1, feedbackPin1, recordButtonPin);
19	}
20	while (b!=0){
21	b= digitalRead(pushButtonPin2);
22	if(b==0) break;
23	recordServo2(servoPin2, feedbackPin2, recordButtonPin);
24	}
25	while (c!=0){
26	c= digitalRead(pushButtonPin3);
27	if(c==0) break;
28	recordServo3(servoPin3, feedbackPin3, recordButtonPin);
29	}
30	while (d!=0){
31	d= digitalRead(pushButtonPin4);
32	if(d==0) break;
33	recordServo4(servoPin4, feedbackPin4, recordButtonPin);
34	}
35	while (e!=0){
36	e= digitalRead(pushButtonPin5);
37	if(e==0) break;
38	recordServo5(servoPin5, feedbackPin5, recordButtonPin);
39	}

5.3.2.4 Kode Program Bergerak Sekali

Kode program bergerak sekali akan berjalan ketika *toggle switch* y mendapatkan nilai 1, lalu menjalankan fungsi pembacaan kembali pada nilai y. dan berikutnya masuk permisalan jika nilai sekali 0 maka akan lanjut menuju pemanggilan method *play*. Kode program berikut ini masuk kedalam method void loop. Dapat dilihat pada Tabel 5.6 berikut.

Tabel 5.6 Kode Program Bergerak Sekali

1	else if(y==1){
2	delay(10);
3	while (y!=0){
4	y= digitalRead(playButtonPin);
5	if(sekali==0){
6	sekali=1;
7	Serial.println("play sekali");
8	playServo1(servoPin1, playButtonPin);
9	playServo2(servoPin2, playButtonPin);
10	playServo3(servoPin3, playButtonPin);
11	playServo4(servoPin4, playButtonPin);
12	playServo5(servoPin5, playButtonPin);
13	}else break;

5.3.2.5 Kode Program Bergerak Berkali-kali

Kode program bergerak berkali-kali akan berjalan ketika *toggle switch* z mendapatkan nilai 1, lalu menjalankan fungsi permissalan ketika dan melakukan pembacaan ke nilai z, setelah itu memanggil method *play*. Fungsi ini akan otomatis berhenti ketika nilai z bernilai 0 atau *toggle switch* bernilai 0. Kode program berikut ini masuk kedalam method void loop. Dapat dilihat pada Tabel 5.7 berikut.

Tabel 5.7 Kode Program Bergerak Berkali-kali

1	else if(z==1){
2	Serial.println("play berkali-kali");
3	delay(10);
4	while (z!=0){
5	z= digitalRead(playButton1Pin);
6	playServo1(servoPin1, playButton1Pin);
7	playServo2(servoPin2, playButton1Pin);
8	playServo3(servoPin3, playButton1Pin);
9	playServo4(servoPin4, playButton1Pin);
10	playServo5(servoPin5, playButton1Pin);

5.3.2.6 Kode Program Inisialisasi Nilai Sekali

Fungsi kode program berikut ini adalah untuk membantu berjalannya bergerak sekali. Maka dikondisikan ketika semua *toggle switch* bernilai 0 maka nilai sekali akan menjadi 0. Kode program berikut ini masuk kedalam method void loop. Sepeti pada Tabel 5.8 berikut.

Tabel 5.8 Kode Program Inisialisasi Nilai Sekali

1	else if(y==0 and x==0 and z==0 and aa==0){
2	sekali=0;

5.3.2.7 Kode Program Record

Fungsi kode *record* merupakan method sendiri. Fungsi ini akan terpanggil di kode program merekam. Ditentukan bahwa nilai alamat pertama yaitu 0. Lalu masuk ke pembacaan pinMode yaitu pembacaan analogPin disini penulis hanya menggunakan fungsi *record* pada servo 1, servo 2 hingga servo 5 menggunakan method yang sama, yang berbeda hanya dibagian pin. Ketika a kurang dari

kalibrasi minimal maka nilai *a* menjadi nilai kalibrasi minimal, sama seperti kalibrasi maksimal. Selanjutnya nilai *a* menjadi nilai *mapping* dan akan ditulis di EEPROM nilai alamat dan nilai *mapping* sebelumnya. Dapat dilihat pada Tabel 5.9 berikut.

Tabel 5.9 Kode Program Record

```

1 void recordServo1(uint8_t servoPin1, uint8_t analogPin1,
2 uint8_t buttonPin) {
3     uint16_t addr = 0;
4
5     Serial.println("Recording");
6     digitalWrite(ledPin, HIGH);
7     pinMode(analogPin1, INPUT);
8     while (digitalRead(buttonPin)) {
9         uint16_t ac = analogRead(analogPin1);
10
11         Serial.print("Merekam Servo 1: "); Serial.print(ac);
12         if (ac < CALIB_MIN) ac = CALIB_MIN;
13         if (ac > CALIB_MAX) ac = CALIB_MAX;
14         ac = map(ac, CALIB_MIN, CALIB_MAX, 0, 254);
15         Serial.print(" -> "); Serial.println(ac);
16         EEPROM.write(addr, ac);
17         if (var1==1) {
18             reset[0][0]=addr;
19             reset[0][1]=ac;
20         }
21         addr++;
22         if (addr == 50) break;
23         delay(100);
24     }
25     if (addr != 25) EEPROM.write(addr, 255);
26
27     digitalWrite(ledPin, LOW);
28     Serial.println("Done");
29
30     delay(2000);
31 }
32

```

5.3.2.8 Kode Program Play

Fungsi kode *play* hampir sama dengan fungsi kode *record*. Di fungsi ini membaca nilai yang sudah disimpan pada EEPROM sebelumnya dan akan menuliskan pada servo berupa nilai *mapping* EEPROM. Dapat dilihat pada Tabel 5.10 berikut.

Tabel 5.10 Kode Program Play

```

1 void playServo1(uint8_t servoPin1, uint8_t buttonPin) {
2     uint16_t addr = 0;
3     Serial.println("Playing");
4
5     myServo1.attach(servoPin1);
6     while (digitalRead(buttonPin)) {
7         uint8_t x = EEPROM.read(addr);
8         Serial.print("Read EE1: "); Serial.print(x);
9         if (x == 25) break;

```

```

10 // map to 0-180 degrees
11 x = map(x, 0, 254, 0, 180);
12 Serial.print(" -> "); Serial.println(x);
13 myServo1.write(x);
14 delay(0.1);
15 addr++;
16 if (addr == 75) break;
17 }
18 Serial.println("Done");
19 myServo1.detach();
20 delay(250);
21 }

```

5.3.2.8 Kode Program Reset

Fungsi Kode Program *Reset* merupakan fungsi pengulangan untuk menulis nilai EEPROM berupa nilai terakhir yang dibaca oleh Fungsi *Record*. Fungsi ini akan terus berulang hingga nilai EEPROM yang sudah di alamatkan pada setiap servo terisi dengan nilai pembacaan akhir. Dapat dilihat pada Tabel 5.11.

Tabel 5.11 Kode Program Reset

```

1 else if (aa==1){
2   Serial.println("RESET");
3   delay(1000);
4   Serial.println("proses reset sedang berlangsung");
5   for(int i=0;i<50;i++) EEPROM.write(i,reset[0][1]);
6   for(int i=50;i<100;i++) EEPROM.write(i,reset[1][1]);
7   for(int i=101;i<150;i++) EEPROM.write(i,reset[2][1]);
8   for(int i=151;i<200;i++) EEPROM.write(i,reset[3][1]);
9   for(int i=201;i<250;i++) EEPROM.write(i,reset[4][1]);
10  var1=1,var2=1,var3=1,var4=1,var5=1;
11  Serial.println("RESET SELESAI");
12 }

```

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian serta pengambilan analisis terhadap sistem yang telah diimplementasikan sebelumnya. Tujuan dari dilakukannya pengujian adalah untuk mengetahui apakah seluruh kebutuhan fungsional dan non-fungsional dapat dipenuhi oleh sistem. Proses pengujian dilakukan pada perangkat keras yang digunakan pada penelitian ini.

6.1 Pengujian Ketepatan Pembacaan Servo

Servo SG90 yang digunakan pada lengan pertama dan kedua merupakan Servo dengan putaran 0° - 180° dan pada lengan ketiga, keempat dan kelima menggunakan servo putaran 0° - 360° , kelima servo tersebut merupakan perangkat keras utama dalam sistem yang akan dibuat dikarenakan memiliki fungsi penggerak dari robot. Pengujian servo putaran 0° - 180° dan 0° - 360° dilakukan dengan cara membuktikan nilai ADC pada sudut istimewa benar sesuai dengan nilai ADC yang didapatkan dari multimeter.

6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui apakah nilai yang dihasilkan dari lengan robot sesuai dengan nilai yang didapatkan dari multimeter.

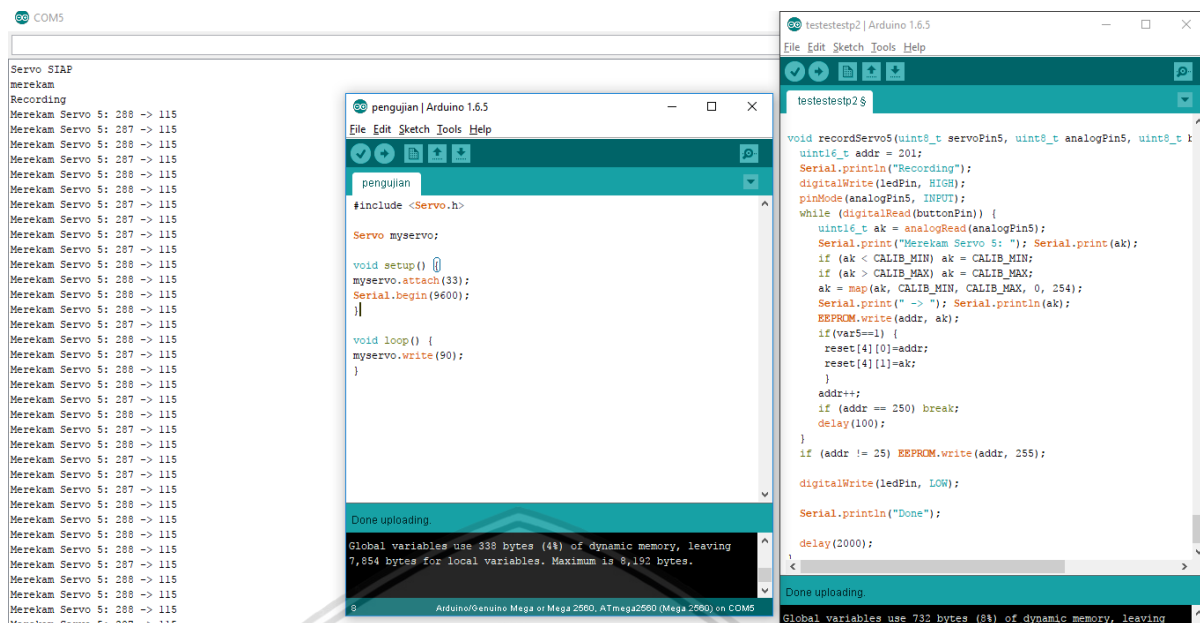
6.1.2 Prosedur Pengujian

Berikut merupakan prosedur yang dilakukan dalam menguji ketepatan pembacaan servo:

1. Servo akan digerakan pada sudut istimewa menggunakan kode program.
2. Hasil sudut istimewa akan mengeluarkan nilai pada pembacaan nilai analog.
3. Servo akan diukur nilai tersebut menggunakan multimeter.
4. Nilai akan dibandingkan dengan multimeter apakah nilai sama dengan menggunakan multimeter.

6.1.3 Hasil dan Analisis Pengujian

Gambar 6.1 berikut ini merupakan hasil ADC yang didapatkan pada servo 0° - 360° pada sudut 90°



Gambar 6.1 Tampilan Pada Serial Monitor

Gambar 6.2 berikut adalah nilai tegangan yang didapatkan pada multimeter



Gambar 6.2 Hasil Pembacaan Pada Multimeter

Untuk mendapatkan nilai ADC yang didapatkan dari multimeter dapat menggunakan rumus *ADC Value to Voltage* :

$$\bullet \frac{\text{Resolution of The ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

- $\frac{1023}{5} = \frac{ADC \text{ Reading}}{Analog \text{ Voltage Measured}}$
- $\frac{1023}{5} \times Analog \text{ Voltage Measured} = ADC \text{ Reading}$

Keterangan : 1. *Analog Voltage Measured* merupakan hasil pembacaan dari multimeter

2. *ADC Reading* merupakan nilai ADC dari hasil nilai voltase.

Nilai dimasukkan kedalam rumus seperti berikut:

- $\frac{1023}{5} \times 1.4v = ADC \text{ Reading}$
- $\frac{1023}{5} \times 1.4v = 286$

Maka didapatkan bahwa nilai ADC pada sudut 180 di servo 5 adalah 286.

Setelah mendapatkan nilai ADC pada multimeter selanjutnya akan dihitung nilai error antara nilai ADC yang dihasilkan oleh servo dan nilai ADC pada multimeter dengan rumus:

$$\text{Nilai error \%} = \frac{(\text{Nilai ADC Servo} - \text{Nilai ADC Multimeter})}{\text{Nilai ADC Servo}} \times 100$$

Berikut merupakan Tabel 6.1 Tabel Pengujian ketepatan pembacaan servo pada servo yang bergerak 0°-180° yang merupakan hasil dari perbandingan nilai ADC yang dihasilkan oleh servo dan oleh multimeter.

Tabel 6.1 Hasil Pengujian Pada Servo 0°-180°

Percobaan Ke-	Servo	Derajat	Nilai ADC Servo	Nilai ADC Multimeter	Error %
1	Servo 0°-180°	0	136	135	0.73 %
2			139	139	0 %
3			136	138	1.47 %
4			135	137	1.48 %
5			137	136	0.72%
6		30	203	194	4.43 %
7			207	199	3.86 %
8			204	200	1.96 %
9			205	201	1.95 %

10	Servo 0°-180°	45	203	197	2.95 %
11			224	225	0.44 %
12			225	225	0 %
13			224	223	0.44 %
14			227	227	0 %
15			225	226	0.44 %
16		60	253	246	2.76 %
17			257	250	2.72 %
18			255	248	2.74 %
19			257	248	3.50 %
20			254	247	2.75 %
21		90	310	327	5.48 %
22			312	322	3.20 %
23			310	325	4.83 %
24			311	326	4.82 %
25			315	322	2.22 %
26		120	367	358	2.45 %
27			369	361	2.16 %
28			368	359	2.44 %
29			365	362	0.82 %
30			366	361	1.36 %
31		135	389	398	2.31 %
32			391	399	2.04 %
33			387	397	2.58 %
34			393	398	1.27 %
35			392	399	1.78 %
36		150	437	430	1.60 %
37			439	433	1.36 %
38			436	434	0.45 %
39			438	431	1.59 %
40			438	435	0.68 %
41		180	474	471	0.63 %
42			477	475	0.41 %

43			475	477	0.42 %
44			478	473	1.04 %
45			477	475	0.41 %
Rata - rata					1.85 %

Dari tabel tersebut merupakan hasil dari pembacaan nilai analog servo serta nilai analog multimeter ketika keadaan servo berada pada posisi sudut istimewa. Posisi sudut tersebut didapatkan dengan cara memprogram servo agar berada pada sudut tersebut setelah itu melakukan analisa hasil. Dapat diketahui nilai rata-rata error adalah 1.85 %.

Tabel 6.2 berikut ini adalah hasil dari pengujian pembacaan dari servo yang bergerak 0° - 360° .

Tabel 6.2 Tabel Pengujian Pada Servo 0° - 360°

Percobaan Ke-	Servo	Derajat	Analog Servo	Analog Multimeter	Error %
1	Servo 0° - 360°	0	-	-	-
2			-	-	-
3			-	-	-
4			-	-	-
5			-	-	-
6		30	158	164	3.79 %
7			160	163	1.87 %
8			161	161	0 %
9			159	162	1.88 %
10			158	163	3.16 %
11		45	196	184	6.12 %
12			198	187	5.55 %
13			195	186	4.61 %
14			197	183	7.10 %
15			195	185	5.12 %
16		60	222	225	1.35 %
17			223	227	1.79 %
18			221	224	1.35 %
19			225	226	0.44 %

20	Servo 0°-360°		224	228	1.78 %
21		90	285	286	0.35 %
22			285	285	0 %
23			283	286	1.06 %
24			288	284	1.38 %
25			284	287	1.05 %
26			120	346	358
27		345		355	2.89 %
28		347		356	2.59 %
29		344		357	3.77 %
30		346		359	3.75 %
31		135		370	378
32			373	380	1.87 %
33			371	379	2.15 %
34			374	381	1.87 %
35			372	377	1.34 %
36			150	401	409
37		402		411	2.23 %
38		401		407	1.49 %
39		400		410	2.50 %
40		403		409	1.48 %
41		180		457	450
42			459	451	1.74 %
43			460	452	1.73 %
44			456	449	1.53 %
45			458	453	1.09 %
Rata - rata					2.32 %

Pada tabel tersebut adalah hasil dari pembacaan nilai analog servo 0°-360° serta nilai ADC multimeter saat servo berada diposisi sudut istimewa. Posisi sudut tersebut didapatkan dengan cara memprogram servo agar berada pada sudut tersebut setelah itu dilakukan analisa. Servo 0°-360° tidak dapat menentukan titik 0°, jika diprogram untuk berada pada 0° maka servo akan terus berputar tanpa henti. Dapat dilihat bahwa nilai rata-rata error adalah 2.32 %.

6.2 Pengujian Perintah 1 Gerakan

Setelah dilakukan pengujian terkait ketepatan servo, selanjutnya dilakukan pengujian perintah 1 gerakan. Pengujian ini dilakukan untuk memastikan bahwa nilai yang dibaca oleh servo ketika diberi sudut belajar sama seperti yang digerakkan oleh servo berupa sudut actual diukur menggunakan busur.

6.2.1 Tujuan Pengujian

ujuan dilakukannya pengujian ini adalah untuk mengetahui apakah gerakan setiap servo yang telah diberi inisialisasi dan diukur menggunakan busur ketika dijalankan bergerak sesuai dengan nilai aktual yang nantinya akan diukur menggunakan busur.

6.2.2 Prosedur Pengujian

Berikut merupakan prosedur yang dilakukan dalam menguji gerakan 1 servo:

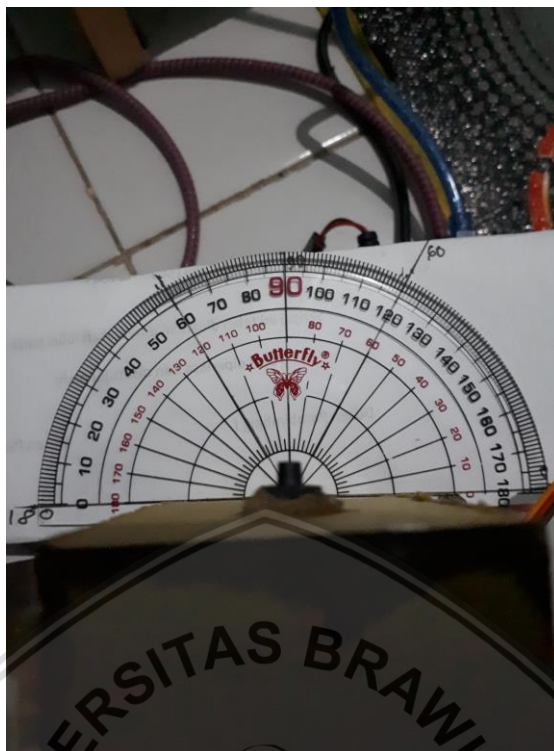
1. Servo akan ditentukan posisi awal pada 90° .
2. Servo akan diberikan 1 gerakan searah.
3. Titik akhir gerakan tersebut akan diukur menggunakan busur.
4. Servo akan diposisikan kembali pada posisi 90° .
5. Menjalankan program bergerak sekali.
6. Mengukur hasil gerakan tersebut dengan busur, apakah nilainya sesuai dengan nilai awal yang sudah diberikan.

6.2.3 Hasil dan Analisis Pengujian

Penempatan posisi servo dilakukan dengan menjalankan kode program `myservo.write` agar posisi servo tepat pada posisi.

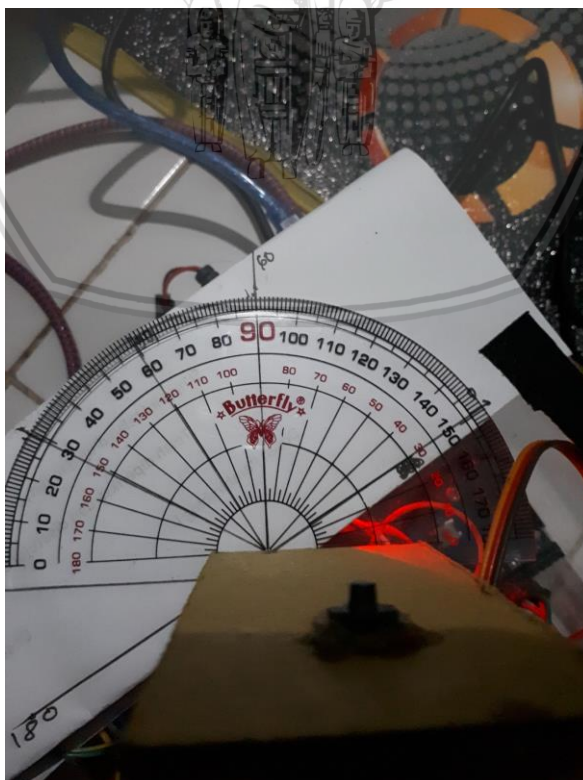
Sama seperti lengan 1, setiap lengan dilakukan pemosisian lengan menggunakan kode program yang sama dan diukur dengan busur. Pemosisian busur ini dilakukan agar nilai sudut belajar yang diberikan dapat dibandingkan dengan nilai sudut yang dihasilkan dan dapat dianalisis nilai error yang dihasilkan.

Pada gambar 6.3 berikut ini merupakan penempatan posisi awal servo pada sudut 90° dan diukur menggunakan busur.



Gambar 6.3 Posisi Awal Servo

Setelah posisi awal servo ditentukan maka servo akan digerakan ke sudut 60° dan merekam gerakan dari 90° menuju 60° . Nilai sudut 60° ini menjadi nilai sudut belajar. Seperti pada gambar 6.4.



Gambar 6.4 Nilai Sudut Belajar

Setelah nilai sudut belajar disimpan oleh EEPROM, maka lengan akan dikembalikan ke sudut 90° dalam keadaan sistem tidak merekam. Lalu program bergerak sekali dijalankan, kemudian diukur menggunakan busur. Didapatkan hasil pada Gambar 6.5.



Gambar 6.5 Hasil Nilai Sudut Aktual

Setelah dilakukan pengujian seperti diatas maka hasil pengujian didapatkan total sudut aktual dan untuk menghitung nilai error digunakan rumus

$$\text{Nilai error \%} = \frac{(\text{Sudut Belajar} - \text{Sudut Aktual})}{\text{Sudut Belajar}} \times 100$$

Tabel 6.3 berikut merupakan hasil pengujian perintah 1 gerakan.

Tabel 6.3 Hasil Pengujian Perintah 1 Gerakan

Pengujian Ke-	Servo	Sudut Awal	Sudut Belajar	Sudut Aktual	Error %
1	Servo 1	90°	45°	45°	0 %
2		90°	45°	45°	0 %
3		90°	60°	63°	5 %
4		90°	60°	63°	5 %

5		90 ^o	120 ^o	120 ^o	0 %
6	Servo 2	90 ^o	45 ^o	47 ^o	4.44 %
7		90 ^o	45 ^o	46 ^o	2.22 %
8		90 ^o	60 ^o	65 ^o	8.33 %
9		90 ^o	60 ^o	64 ^o	6.67%
10		90 ^o	120 ^o	122 ^o	1.67 %
11	Servo 3	90 ^o	45 ^o	45 ^o	0 %
12		90 ^o	45 ^o	44 ^o	2.22 %
13		90 ^o	60 ^o	62 ^o	3.33 %
14		90 ^o	60 ^o	61 ^o	1.67 %
15		90 ^o	120 ^o	124 ^o	3.33 %
16	Servo 4	90 ^o	45 ^o	45 ^o	0 %
17		90 ^o	45 ^o	44 ^o	2.22 %
18		90 ^o	60 ^o	63 ^o	5 %
19		90 ^o	60 ^o	61 ^o	1.67 %
20		90 ^o	120 ^o	121 ^o	0.83 %
21	Servo 5	90 ^o	45 ^o	41 ^o	8.89 %
22		90 ^o	45 ^o	42 ^o	6.67 %
23		90 ^o	60 ^o	61 ^o	1.67 %
24		90 ^o	60 ^o	60 ^o	0 %
25		90 ^o	120 ^o	122 ^o	1.67 %
Rata-rata					2.9 %

Dari tabel tersebut didapatkan hasil dari pembacaan perintah 1 gerakan dengan posisi awal semua servo 90° lalu digerakan dengan beberapa sudut istimewa. Posisi servo awal ditentukan dengan cara mengisialisasi pada program, setelah itu dilakukan penempatan posisi busur dan dilakukan fungsi merekam pada servo sesuai dengan derajat pada busur. Setelah perubahan sudut direkam, lengan akan dikembalikan pada posisi 90°, selanjutnya menjalankan program bergerak sekali dan dibandingkan posisi lengan yang sudah direkam sebelumnya dengan nilai aktual. Dapat diketahui nilai rata-rata error adalah 2.9 %.

6.3 Pengujian Perintah Keseluruhan

Setelah melakukan pengujian perintah 1 gerakan pada masing-masing servo selanjutnya dilakukan pengujian perintah keseluruhan, pengujian perintah

keseluruhan ini dilakukan dengan mengukur gerakan servo 1 hingga 5 dalam satu rekaman apakah sesuai dengan yang dijalankan, lalu diukur dengan menggunakan busur.

6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui apakah gerakan keseluruhan servo yang telah diberi inisialisasi dan diukur menggunakan busur ketika dijalankan bergerak sesuai dengan nilai aktual yang nantinya akan diukur menggunakan busur.

6.3.2 Prosedur Pengujian

Berikut merupakan prosedur yang dilakukan dalam menguji ketepatan pembacaan servo:

1. Setiap servo akan ditentukan posisi awal pada 90° .
2. Setiap servo akan diberikan 1 gerakan searah.
3. Titik akhir gerakan tersebut akan diukur menggunakan busur.
4. Servo akan diposisikan kembali pada posisi 90° .
5. Menjalankan program bergerak sekali.
6. Mengukur hasil gerakan tersebut dengan busur, apakah nilainya sesuai dengan nilai awal yang sudah diberikan.

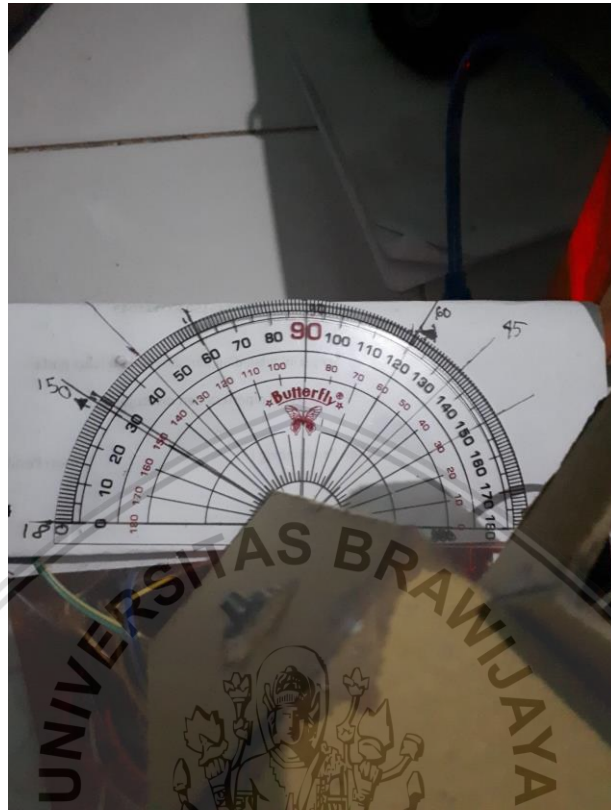
6.3.3 Hasil dan Analisis Pengujian

Berikut ini merupakan Gambar 6.6 kondisi salah satu lengan/servo setelah merekam hasil gerakan menuju sudut 150° dan dikembalikan pada posisi 90° .



Gambar 6.6 Posisi Servo Setelah Dilakukan Perekaman

Pada Gambar 6.7 adalah salah satu hasil pergerakan servo yang sedang diukur perbandingan sudut belajar dengan sudut aktual.



Gambar 6.7 Hasil Sudut Aktual

Pengujian dilakukan setiap kondisi bergerak dengan sudut yang sama. Maka didapatkan hasil seperti pada Tabel 6.4.

Tabel 6.4 Hasil Pengujian Perintah Keseluruhan

Pengujian Ke-	Servo	Sudut Awal	Sudut Belajar	Sudut Aktual	Error %
1	Servo 1	90°	30°	31°	3.33 %
	Servo 2	90°	30°	30°	0 %
	Servo 3	90°	30°	32°	6.67 %
	Servo 4	90°	30°	31°	3.33 %
	Servo 5	90°	30°	31°	3.33 %
2	Servo 1	90°	45°	45°	0 %
	Servo 2	90°	45°	46°	2.22 %
	Servo 3	90°	45°	45°	0 %
	Servo 4	90°	45°	44°	2.22 %
	Servo 5	90°	45°	46°	2.22 %

3	Servo 1	90°	60°	62°	3.33 %
	Servo 2	90°	60°	60°	0 %
	Servo 3	90°	60°	60°	0 %
	Servo 4	90°	60°	61°	1.67 %
	Servo 5	90°	60°	62°	3.33 %
4	Servo 1	90°	120°	118°	1.67 %
	Servo 2	90°	120°	121°	0.83 %
	Servo 3	90°	120°	120°	0 %
	Servo 4	90°	120°	122°	1.67 %
	Servo 5	90°	120°	120°	0 %
5	Servo 1	90°	150°	153°	2 %
	Servo 2	90°	150°	152°	1.33 %
	Servo 3	90°	150°	150°	0 %
	Servo 4	90°	150°	151°	0.67 %
	Servo 5	90°	150°	150°	0 %
Rata-rata					1.5 %

Dari tabel tersebut didapatkan hasil dari pembacaan perintah keseluruhan dengan posisi awal semua servo 90° lalu disetiap percobaan digerakan dengan beberapa sudut istimewa. Posisi servo awal ditentukan dengan cara mengisialisasi pada program, setelah itu dilakukan penempatan posisi busur dan dilakukan fungsi merekam pada setiap servo sesuai dengan derajat pada busur. Setelah perubahan sudut direkam, lengan akan dikembalikan pada posisi 90°, selanjutnya menjalankan program bergerak sekali dan dibandingkan posisi lengan yang sudah direkam sebelumnya dengan nilai aktual. Dapat diketahui nilai rata-rata error adalah 1.5 %.

BAB 7 PENUTUP

Pada bab ini akan diberikan kesimpulan yang didapat setelah melakukan proses perancangan, implementasi, pengujian serta analisis pada sistem. Kemudian diberikan pula saran yang harapannya dapat dipertimbangkan untuk pengembangan sistem yang serupa kedepannya.

7.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dibuat pada awal penelitian serta pengujian dan analisis yang telah dilakukan, maka dapat diambil beberapa kesimpulan, diantaranya:

1. Untuk merancang gerak robot berdasarkan perintah inisialisasi awal pengguna dibutuhkan perangkat keras berupa servo motor, arduino mega2560, *push button*, *toggle switch*, rangkaian *debounce* serta perangkat laptop dan perangkat lunak yaitu arduino.
2. Implementasi kendali robot berdasarkan perintah inisialisasi awal pengguna dapat dilakukan setelah melakukan rancangan sistem, implementasi yang dilakukan pertama adalah dengan membuat rangkaian *debounce* dilanjutkan dengan mencari *feedback* yang ada di servo, lalu pembuatan lengan robot, sebelumnya peneliti menggunakan bahan arkilik dalam pembuatan lengan robot, lalu diganti menggunakan karton duplex. Dilanjutkan dengan implementasi penampang dan pemasangan kabel.
3. Analisis hasil gerak servo pada kendali robot berdasarkan perintah inisialisasi awal pengguna ini dapat dihasilkan bahwa nilai ADC yang dihasilkan hampir serupa dengan nilai ADC yang diukur menggunakan multimeter. Dapat dilihat nilai rata-rata error hanya sebesar 1.85 % dan 2.32 %. Lalu pada analisis perintah 1 gerakan nilai rata-rata error hanya sebesar 2.9 % dan pada analisis perintah keseluruhan nilai rata-rata error hanya sebesar 1.5 %.

7.2 Saran

Adapun beberapa saran yang dapat penulis berikan untuk pengembangan sistem kedepannya, adalah:

1. Menggunakan servo yang lebih kuat dan kokoh dalam menumpu beban yang lebih berat dari total bahan dan servo itu sendiri dikarenakan dengan menggunakan servo sg90 beban yang bisa diterima hanya kecil.
2. Menggunakan bahan yang lebih kuat dari bahan sebelumnya, dengan menggunakan bahan yang lebih kuat maka robot akan lebih kokoh dikarenakan sebelumnya peneliti menggunakan akrilik namun servo tidak kuat bergerak karena beban lebih berat dari beban yang seharusnya dapat diterima.
3. Menggunakan fungsi array dikarenakan fungsi EEPROM hanya dapat membaca sedikit data dan dibagi berdasarkan servo, maka dengan menggunakan fungsi array maka dapat menyimpan data analog yang lebih banyak.

4. Mengurangi penggunaan kabel dikarenakan penggunaan kabel dalam jumlah yang banyak dapat mempengaruhi beban berat dari robot.
5. Menggunakan LCD agar tampilan lebih mudah dilihat.



DAFTAR PUSTAKA

- Aditya Hidayat, A. N. J. a. R. E. S., 2016. Autonomous Quadruped Robot Locomotion Control Using Inverse Kinematics and Sine Pattern Methods. *The 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*.
- Biorobotics Laboratory, E., 2015. *BIROBOTICS LABORATORY BIOROB*. [Online] Available at: <http://biorob.epfl.ch/cheetah> [Accessed 3 March 2017].
- Biswas Palok, S. A. S., 2016. Design and Development of a 3 axes Pneumatic Robotic Arm. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, V(9).
- Department of Labor, U. S., 2017. *Occupational Safety and Health Administration*. [Online] Available at: https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html [Accessed 23 September 2017].
- Dr. Bindu A Thomas, S. M. S. V. B. N. S. M., 2013. Industry Based Automatic Robotic Arm. *International Journal of Engineering and Innovative Technology (IJEIT)*, II(11).
- Gonzales, C. M., 2016. What's the differences between industrial robots?. *machine design*.
- Jinrong Zhang, C. W. a. J. Z., 2014. The Kinematics Analysis and Configuration Optimize of Quadruped Robot. *The Open Automation and Control Systems Journal*.
- K. Weinmeister¹, P. E. H. W. a. A.-J. I., 2015. Cheetah-cub-S: Steering of a Quadruped Robot using Trunk Motion.
- KOLLMORGEN, 2016. *Motion Control Solution*. [Online] Available at: <https://www.kollmorgen.com/en-us/products/motors/servo/servo-motors/> [Accessed September 2017].
- Liu, C.-Y. C. a. Y.-C., 2016. Towards a Walking, Turning, and Jumping Quadruped Robot with Compliant Mechanisms. *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*.
- Massimo Banzi, D. C. T. I. G. M. a. D. M., 2017. *Arduino*. [Online] Available at: arduino.cc
- Md. Anisur Rahman, A. H. K. D. T. A. d. M. M. S., 2013. Design, Analysis and Implementation of a Robotic Arm- The Animator. *American Journal of Engineering Research (AJER)*, II(10).

Narwane, S. O. a. V., 2014. Design, Analysis and Fabrication of Quadruped Robot with Four bar Chain Leg Mechanism. *IJSET - International Journal of Innovative Science, Engineering & Technology*, 1(6).

Park, B. T., 2014. *Immersa Lab.* [Online] Available at: <http://www.immersa-lab.com/> [Accessed December 2017].

Putra, D. A. E., 2009. *DSP & Embedded Electronics.* [Online] Available at: agfi.staff.ugm.ac.id [Accessed 2017].

Ryosuke Kawasaki, R. S. E. K. A. M. a. M. S., 2016. Development of A Flexible Coupled Spine Mechanism For A Small Quadruped Robot. *International Conference on Robotics and Biomimetics*.

Satoshi Nishikawa¹, K. S. a. Y. K., 2016. Musculoskeletal Quadruped Robot with Torque-Angle Relationship Control System. *2016 IEEE International Conference on Robotics and Automation (ICRA)*.

Thing, L., 2016. *TechTarget WhatIs?* [Online] Available at: <http://whatis.techtarget.com/definition/initialization> [Accessed December 2017].

Thing, L., 2017. *TechTarget Whatis.com?* [Online] Available at: <http://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory> [Accessed December 2017].